

# Improving the Detection of Gravitational-Wave Signals in Real Time

Arthur Tolley



UNIVERSITY OF PORTSMOUTH

October 2024

The thesis is submitted in partial fulfilment of the requirements for the award of the degree of

*Doctor of Philosophy*

of the

UNIVERSITY OF PORTSMOUTH



*“Yesterday is history, tomorrow is a mystery, but today is a gift—that’s why it is called the present.”*

— Grand Master Oogway



# Declaration of Authorship

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

The contents of this thesis were examined in a viva voce held on 3<sup>rd</sup> December 2024 by **Prof Graham Woan** and **Prof Kazuya Koyama**, to whom special thanks are owed. All research underlying this thesis has passed ethics review at the University of Portsmouth with certification code: ETHICS-10374.

This thesis has an estimated word count of **42,069**, this is calculated by the Overleaf word count tool. This thesis was originally submitted on the 30<sup>th</sup> September 2024 by **Arthur Tolley**.



This work by Arthur Tolley is licensed under  
**Creative Commons Attribution 4.0 International.**  
(<https://creativecommons.org/licenses/by/4.0/>).



# Abstract

This thesis presents advancements in the detection of gravitational waves from compact binary coalescences, utilising the most sensitive observatories constructed to date. The research focuses on enhancing gravitational-wave signal searches through the development of new tools and the application of existing methodologies to increase the sensitivity of live gravitational-wave searches.

We introduced a novel noise artefact model, which enabled the identification and removal of glitches, thereby facilitating the recovery of previously missed gravitational-wave injections. This pioneering approach established a glitch search pipeline that adapted techniques typically used in gravitational-wave searches to address the unique characteristics of glitches. Additionally, we implemented an exponential noise model within the PyCBC Live search framework, significantly improving the detection ranking statistics for gravitational-wave signals and demonstrating the potential for substantial increases in detection sensitivity.

Furthermore, we analysed and proposed enhancements for the PyCBC Live Early Warning search to maximise the detection of gravitational-wave events in the early warning regime. Our findings highlighted deficiencies in the current ranking statistic and led to recommendations for optimising coincidence timing windows and refining phase-time-amplitude histograms. These adjustments aim to increase the detection of gravitational-wave signals, particularly binary neutron star events, in early warning scenarios.

The results underscore the importance of advancing search techniques in gravitational-wave astronomy, which can operate independently of detector improvements. By refining search methodologies, we enhance the capacity to detect a greater number of events, contributing significantly to our understanding of the Universe.



# Acknowledgements

Firstly, I owe special thanks to Graham and Kazuya for examining my thesis and conducting my viva voce. Thank you to my academia supervisors: Andy Lundgren, you gave me the first opportunity to perform some ‘proper’ research; Ian Harry, you supported me throughout the PhD as my main supervisor, ensuring I was suitably challenged and consequently over-prepared all the way to the end; Gareth and Xan you have both helped me immeasurably with my projects and saved me from countless hours of debugging.

I required a great deal of personal support during the PhD, and for that the bulk of thanks goes to my amazing partner, Vickie. Without you by my side, I’m sure I wouldn’t have made it to the end. There were many moments I was ready to quit and get a job that paid me a living wage. Thankfully, you were there to help me out and maintain my lavish lifestyle, I love you forever.

To my friends in Ipswich: Billy, George, Ricky, Muntazim, I am thankful for the friendship we’ve shared throughout primary school, high school and sixth form. To the friends I made during my undergraduate degree: Chris, Akash, Nick and Alex, we’ve had great adventures and living together made University feel like a breeze, I truly wish I could experience it all again. I want to say special thanks to Chris for being an amazing flatmate during the initial years of the PhD, I miss the days when we would spend hours chatting in the kitchen.

Thanks to the many PhD students I have shared a department with, you made this journey far less daunting than it could have been, foremost are my cohort: Connor, Rafaela, Dan, and Joe. Connor, we spent hours ranting about the PhD, you were the best support a man could’ve asked for and knowing we weren’t alone in how we felt meant we both made it to the end, and it wasn’t all for nothing. Love you buddy.

The root cause of my success is my family. These people have moulded me into the person I am, and I love you all more than anything. Jack and India, thank you for the childhood we shared. Mum and Lee, thank you for your constant support and safety net. Dad, I am thankful for who you are today and I know you’re always trying your best, Jo, thank you for forcing him to try his best! Claire, thank you for always doing the best for me, no matter the circumstances.



# **Contents**



# Introduction

In the 20th century, Albert Einstein introduced the theory of General Relativity [? ? ? ? ? ?], which describes gravity as the curvature of spacetime caused by massive objects, rather than as a force. In this framework, objects move along geodesics in curved spacetime. A key prediction of the theory is the existence of gravitational waves [? ? ? ? ? ?], which are perturbations in spacetime generated by the acceleration and collisions of massive bodies, such as black holes or neutron stars. These waves propagate at the speed of light and induce small changes in the proper distance between freely falling objects. The detector strain caused by gravitational waves is extremely small, typically on the order of  $10^{-22}$  for the most energetic sources, making direct detection highly challenging. Despite this, the observation of gravitational waves has become a crucial tool for studying astrophysical phenomena and testing the predictions of General Relativity.

The first direct detection of gravitational waves occurred in 2015 with the observation of GW150914 [?], a gravitational-wave signal produced by the merger of two black holes. This event was detected by the Laser Interferometer Gravitational-Wave Observatory (LIGO) and the Virgo [?] gravitational-wave observatory using ground-based interferometers. The observed signal matched the predictions of General Relativity [?] for the inspiral, merger, and ringdown phases of a binary black hole coalescence. This detection marked the beginning of gravitational-wave astronomy, providing the first direct evidence of black hole mergers and opening a new observational window to study the Universe.

Following the first detection of gravitational waves, the global gravitational-wave detector network expanded with the inclusion of the KAGRA detector [? ]. The current International Gravitational-Wave Observatory Network (IGWN) collaboration has enabled more comprehensive and sensitive observations of gravitational-wave events. With a running total of 221 gravitational-wave events [? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ] from compact binary coalescences, including two mergers of binary neutron star systems [? ? ] and two neutron star–binary black hole mergers [? ], across four observing runs as of the drafting of this thesis.

The work performed in this thesis focuses on improvements to the current search



methods for gravitational waves from data obtained by ground-based gravitational-wave observatories. Chapter ?? provides a simple derivation of gravitational waves, a brief discussion of how the current generation of detectors operates, and the expected gravitational-wave emissions from a compact binary coalescence. This discussion is intended to give the reader a basic understanding of the background required to appreciate the motivation for the research conducted in later chapters; for texts with more detailed discussions, please refer to [? ? ? ? ]. Chapter ?? reviews the detection methods used for gravitational waves, focusing on the PyCBC [?] search pipeline. Chapter ?? concentrates on Detector Characterisation and the techniques employed to characterise and calibrate ground-based interferometer noise, producing high-quality data for further study.

Chapter ?? investigates the feasibility of modelling gravitational-wave detector data artefacts, specifically the scattered-light artefact, and subtracting these artefacts to clean the data before the search for gravitational waves. In Chapter ??, we describe the improvements to the PyCBC Live search’s ranking statistic aimed at enhancing detection sensitivity. Chapter ?? details the PyCBC Live Early Warning search, examining its deficiencies and suggesting improvements to ensure the identification of all potentially electromagnetically bright events. In Chapter ??, we discuss the investigation and enhancements made to the PyCBC Live rapid signal-to-noise ratio optimiser. Finally, in Chapter ??, we briefly discuss the industrial placements undertaken as a requirement of this doctoral programme.

# 1

---

## Gravitational Waves

---

*“This little manoeuvre’s gonna cost us 51 years.”*

— Joseph Cooper, *Interstellar*

In this thesis, we focus on improving the capabilities of gravitational-wave search pipelines to detect gravitational-wave signals from compact binary coalescences (CBCs). As an introduction to the research chapters later in this thesis, we will briefly discuss gravitational radiation, its detection using laser interferometry and, the gravitational-wave signals we expect to see from CBCs. In this chapter, we construct a simple description of gravitational waves as a consequence of General Relativity. In Section ?? we derive gravitational waves as solutions of the Einstein field equations in a linearised gravity regime under the weak-field approximation. In Section ?? we will discuss the ground-based gravitational-wave observatories and how gravitational waves will create physical effects that we can measure using these observatories. Finally, in Section ??, we model the gravitational waveform from the primary source of gravitational waves—CBCs—we see with our detectors. For greater detail on any of these sections, readers are referred to the texts [?? ?? ].

### 1.1 Gravitational radiation

Gravitational radiation refers to the propagation of perturbations in spacetime, predicted by General Relativity, that manifest as gravitational waves. These waves are



generated by the accelerated motion of massive objects and propagate at the speed of light. In this section, we will derive gravitational waves as solutions to the Einstein field equations within the weak-field approximation, where the gravitational field is considered a small perturbation to flat spacetime. This approach, known as linearised gravity, allows for a simplified treatment of gravitational waves and provides the foundation for understanding their key properties. We will also discuss the general characteristics of gravitational radiation, such as their polarisation and energy carried by gravitational waves.

Einstein's field equations (commonly referred to as Einstein's equations) are a set of 10<sup>1</sup> non-linear coupled equations which describe how the gravitational field is generated by matter and can be written eloquently using General Relativity notation as

$$G_{\mu\nu} = 8\pi T_{\mu\nu}, \quad (1.1)$$

where we use natural units,  $G = c = 1$ .

Einstein's equations relate the curvature of spacetime using the Einstein tensor,  $G_{\mu\nu}$ , to the stress-energy tensor,  $T_{\mu\nu}$ , which describes the density and flux of matter, energy, and momentum at each point in spacetime.

### 1.1.1 Perturbations in spacetime

Gravitational waves are a natural consequence of General Relativity and we show this by making a number of simplifications to Einstein's equations to reveal valid plane wave solutions. We begin in a flat-Minkowski spacetime, represented by the spacetime metric,  $\eta_{\mu\nu}$ , and make a *weak* linear perturbation,  $h_{\mu\nu}$  due to a gravitational field. In this scenario we can write the metric tensor,  $g_{\mu\nu}$ , as

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad (1.2)$$

and to ensure our perturbation is small we enforce the condition  $|h_{\mu\nu}| \ll 1$ . In this regime, we are able to raise and lower indices using the Minkowski metric

$$h^{\mu\nu} = \eta^{\mu\alpha}\eta^{\nu\beta}h_{\alpha\beta}. \quad (1.3)$$

Terms of  $h_{\mu\nu}$  greater than linear-order are negligible and are ignored, this is called taking the *weak-field limit*. This simplification derives gravitational waves in a regime

---

<sup>1</sup>Due to the symmetry of  $G_{\mu\nu}$  and  $T_{\mu\nu}$ .



where Einstein's equations are linearised, which significantly reduces the complexity of the calculations while still capturing the essential physical behaviour of gravitational waves.

We can write the Einstein tensor,  $G_{\mu\nu}$ , in terms of the Riemann tensor,  $R_{\mu\nu}$ , the Ricci scalar,  $R$ , and the metric tensor

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R, \quad (1.4)$$

and we can introduce a quantity known as the *trace reverse tensor*

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h, \quad (1.5)$$

where  $h = \eta^{\mu\nu}h_{\mu\nu}$  is the trace of the metric perturbation.

We can then express the linearised Einstein tensor in terms of the linearised Riemann tensor and our trace reverse metric tensor [?]

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = \frac{1}{2}\left(\partial^\sigma\partial_\mu\bar{h}_{\sigma\nu} - \partial^\sigma\partial_\sigma\bar{h}_{\mu\nu} + \partial_\nu\partial^\alpha\bar{h}_{\mu\alpha} - \eta_{\mu\nu}\partial^\alpha\partial^\beta\bar{h}_{\alpha\beta}\right). \quad (1.6)$$

We can then express our fully linearised Einstein field equations as

$$-\square\bar{h}_{\mu\nu} + \partial^\sigma\partial_\mu\bar{h}_{\sigma\nu} + \partial_\nu\partial^\alpha\bar{h}_{\mu\alpha} - \eta_{\mu\nu}\partial^\alpha\partial^\beta\bar{h}_{\alpha\beta} = 16\pi T_{\mu\nu}, \quad (1.7)$$

where  $\square$  is the flat space d'Alembertian operator,

$$\square = \partial^\sigma\partial_\sigma = -\frac{\partial^2}{\partial t^2} + \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}. \quad (1.8)$$

### 1.1.2 Gauge transformations

Inspecting Equation ?? we can identify the presence of the d'Alembertian operator  $\square$  along with terms involving divergences of  $\bar{h}_{\mu\nu}$ . Performing a gauge transformation to the harmonic gauge will eliminate the divergences,

$$\partial^\nu\bar{h}_{\mu\nu} = 0, \quad (1.9)$$

further simplifying our linearised Einstein equations.

In the weak-field limit we can make this gauge transformation by applying small coordinate translations to the perturbed spacetime that ensure the metric perturba-



tions remain small,

$$x \rightarrow x', \quad (1.10)$$

$$x'^\mu = x^\mu + \xi^\mu(x), \quad (1.11)$$

where the metric in the new coordinate system is

$$g'_{\mu\nu} = \eta_{\mu\nu} + \bar{h}'_{\mu\nu}, \quad (1.12)$$

and remains valid as long as the weak-field approximation holds in the new coordinates  $|h'_{\mu\nu}| \ll 1$ .

We have chosen a coordinate system where Equation ?? is satisfied, and in this gauge, the Einstein field equations simplify to

$$\square \bar{h}_{\mu\nu} = -16\pi T^{\mu\nu}. \quad (1.13)$$

### 1.1.3 Gravitational waves in vacuum

In the weak-field approximation, where we are far from any sources of mass or energy ( $T_{\mu\nu} = 0$ ), Equation ?? simplifies to

$$\square \bar{h}_{\mu\nu} = 0, \quad (1.14)$$

which is the wave equation and has plane wave solutions for  $\bar{h}_{\mu\nu}$ ,

$$\bar{h}_{\mu\nu} = A_{\mu\nu} e^{i(k_\alpha x^\alpha)}, \quad (1.15)$$

indicating the wave nature of gravitational perturbations.  $A_{\mu\nu}$  is the amplitude tensor and  $k_\alpha = (-\omega, \mathbf{k})$ , the wave vector that satisfies

$$k^\alpha k_\alpha = 0 = -\omega^2 + |\vec{k}|^2. \quad (1.16)$$

Equation ?? tells us that  $\omega^2 = |k|^2$  meaning the speed of gravity  $v = \omega/|k_i| = 1$ : gravitational waves travel at the speed of light.

### 1.1.4 The transverse-traceless gauge

We now have solutions to the Einstein field equations in the form of plane waves.  $\bar{h}_{\mu\nu}$  is a tensor which contains 10 independent components (as a symmetric tensor in



4-dimensional spacetime) with redundancy between them. We can express another set of gauge transformations to reduce this to a tensor with only 4 components and 2 degrees of freedom.

First, we apply the gauge condition stated in Equation ?? to  $\bar{h}_{\mu\nu}$  to obtain,

$$0 = k^\mu A_{\mu\nu}, \quad (1.17)$$

this implies that gravitational waves are *transverse*. This removes the components of  $\bar{h}_{\mu\nu}$  that are parallel to the direction of propagation of the gravitational wave; hence,  $\bar{h}_{0i} = 0$ <sup>2</sup> and  $A_{t\mu} = 0$ . For instance, a gravitational wave travelling in the  $z$ -direction will have purely spatial components in the  $x$  and  $y$  directions.

Second, we require  $\bar{h}_\mu^\mu = 0$ , the trace of the gravitational wave to vanish. This removes the possibility of a volume change in spacetime due to gravitational-wave transmission. We say that our gravitational waves are now *traceless*. This gauge is called the *transverse-traceless* (TT) gauge.

### 1.1.5 The propagation of gravitational waves

In the TT gauge, we can express the metric perturbation  $h_{\mu\nu}^{TT}$  for a gravitational wave propagating in the  $z$ -direction

$$h_{\mu\nu}^{TT} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (1.18)$$

We define the plus and cross polarisations as [? ]

$$h_+ = A_+ \cos(\omega t - \omega z + \phi_0), \quad (1.19)$$

$$h_\times = A_\times \cos(\omega t - \omega z + \phi_0), \quad (1.20)$$

where  $A_+$  and  $A_\times$  are the amplitudes of the two polarisations, and  $\phi_0$  is a constant phase offset. We have described the metric perturbation with only two independent components: the plus (+) and cross ( $\times$ ) polarisations, representing the two degrees of freedom in gravitational radiation.

We consider the effect of a passing gravitational wave on two particles in the  $x$ - $y$  plane placed at  $(x_1, y_1, 0)$  and  $(x_2, y_2, 0)$ . The spacetime interval between the two

---

<sup>2</sup>Roman letter indices represent spatial dimensions ( $x, y, z$ ).

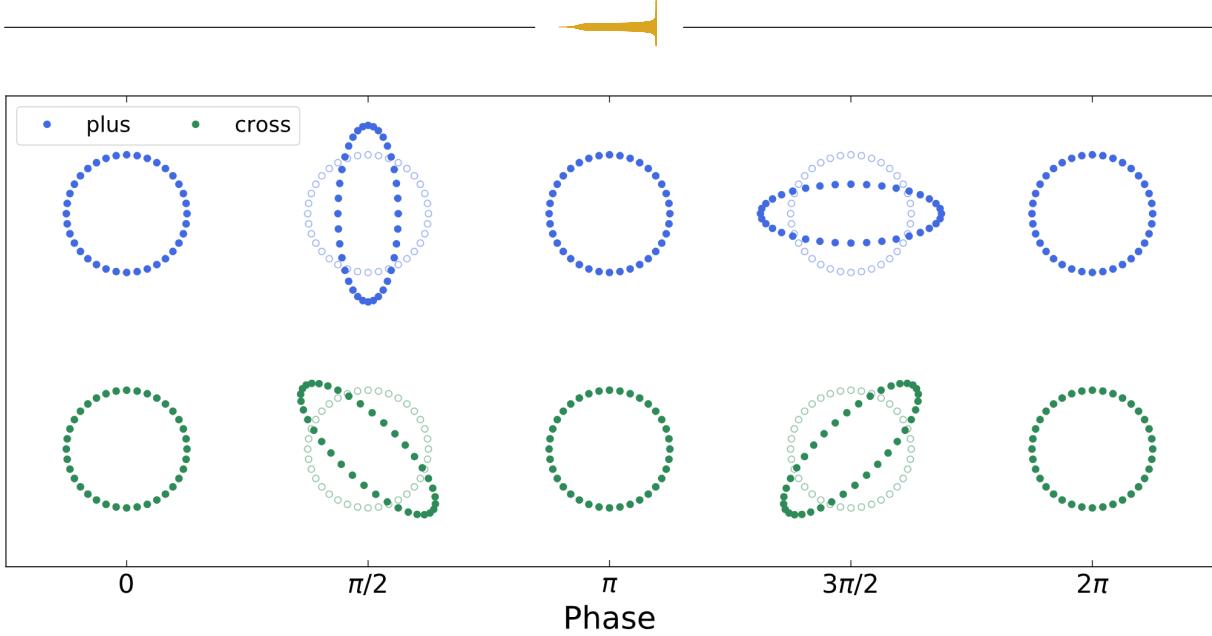


Figure 1.1: The effect of the two polarisations on a ring of test particles [? ].

particles can be written

$$\Delta s^2 = \left( \eta_{\mu\nu} + h_{\mu\nu}^{TT} \right) \Delta x^\mu \Delta x^\nu, \quad (1.21)$$

$$\begin{aligned} &= -dt^2 + (1 + h_+) (x_1 - x_2)^2 \\ &\quad + (1 - h_+) (y_1 - y_2)^2 \\ &\quad + 2h_\times (x_1 - x_2)(y_1 - y_2). \end{aligned} \quad (1.22)$$

and in the simple case where ( $y_2 - y_1 = 0$ ) we can calculate the proper distance between the two particles at an arbitrary time,  $t$ , as

$$\Delta s \approx \left( 1 + \frac{1}{2}h_+ \right) (x_1 - x_2). \quad (1.23)$$

We can see that the proper distance between the two particles is stretched by the factor  $(1 + \frac{1}{2}h_+)$ . It can be shown that the simultaneous motion in the  $y$ -direction results in a squeezing of the proper distances to  $(1 - \frac{1}{2}h_+)$ . This change in proper distance is called gravitational-wave “strain” and is what we want to measure to directly observe the effects of gravitational waves.

Figure ?? provides a visual representation of how the proper distances in a ring of particles change under the influence of a gravitational wave with pure + or  $\times$  polarisation. It can be seen that the effect of the  $\times$  polarisation is the same as the + polarisation but with a  $45^\circ$  rotation.



### 1.1.6 Gravitational-wave emission in linearised gravity

We now return to Equation ?? and reconsider the stress-energy tensor. The solution to the Einstein equations can be obtained using the Green's function [? ],

$$\bar{h}_{\mu\nu}(t, \vec{x}) = 4 \int d^3x' \frac{T_{\mu\nu}(t - |\vec{x} - \vec{x}'|, \vec{x}')}{|\vec{x} - \vec{x}'|}, \quad (1.24)$$

where the observer is at  $\vec{x}$  and the source is at  $\vec{x}'$ . The solution can be approximated using the quadrupole formula under two assumptions: the observer is far away ( $|\vec{x} - \vec{x}'| \approx |\vec{x}| = r$ ) and the source velocity is small compared to the speed of light.

We expand the stress-energy tensor using a Taylor expansion around the retarded time  $(t - r)$ , and Equation ?? simplifies to

$$\bar{h}_{\mu\nu}(t, \vec{x}) = \frac{4}{r} \int d^3x' T_{\mu\nu}(t - r, \vec{x}'), \quad (1.25)$$

to leading order. We can see that the gravitational waves are dependent on the integral of the stress-energy tensor over a volume containing the source, evaluated at the retarded time.

Using the laws of conservation of total energy and momentum,  $\partial^\mu T_{\mu\nu} = 0$ , we can see that the integral of the  $T_{tv}$  and  $T_{vt}$  components will be constant in time. We choose to ignore these components and focus only on observable signals in the spatial dimensions moving forward.

The right-hand side of Equation ?? can be expressed as

$$\frac{4}{r} \int d^3x' T_{ij} = \frac{2}{r} \frac{\partial^2}{\partial t^2} \int T_{00} x_i x_j d^3x', \quad (1.26)$$

where  $T_{00} \approx \rho$ , the Newtonian mass density. This expression is referred to as the *mass quadrupole moment tensor* of the mass distribution

$$M_{ij} := \int T_{00} x_i x_j d^3x', \quad (1.27)$$

and when substituted into Equation ??, we obtain the *quadrupole formula*

$$h_{ij} = \frac{2}{r} \ddot{M}_{ij}(t - r), \quad (1.28)$$

where each dot over a tensor represents on derivative with respect to time.

The key properties of this equation are the dependence of gravitational radiation



on the second time derivative of the quadrupole moment tensor, indicating that gravitational waves are produced by rapid changes in the quadrupole moment tensor, such as the accelerations of masses and their directional changes during orbital cycles. Additionally, the gravitational-wave strain falls off inversely with distance  $r$  from the source.

To extract the plus and cross polarisations of the metric perturbation we must now transform into the transverse-traceless gauge, this is done using the projection tensor [? ]

$$P_j^i(\hat{n}) = \delta_j^i - n^i n_j \quad (1.29)$$

where

$$P_j^i P_k^j = P_k^i \quad (1.30)$$

and  $\hat{n}$  is a normal vector such that  $|\hat{n}| = 1$ . Applying the projection vector to a three-dimensional vector will project it down into the two-dimensional plane orthogonal to  $\hat{n}$ . To get the components of  $h_{ij}$  which are transverse to  $\hat{n}$  we need to apply the projection tensor once for each index

$$h_{kl}^T = P_k^i P_l^j h_{ij}, \quad (1.31)$$

and then to remove the trace of the tensor we use the Minkowski metric

$$h^T = \eta^{ij} h_{ij}^T = P^{ij} h_{ij}. \quad (1.32)$$

From this we can construct the tensor  $\frac{1}{2}P_{kl}h^T$  which will have a trace of  $h^T$  and will be transverse to  $\hat{n}$  allowing us to subtract it from the transverse metric perturbation to get the metric perturbation in its transverse-traceless form

$$h_{kl}^{TT} = \Lambda_{kl}^{ij} h_{ij} \quad (1.33)$$

where

$$\Lambda_{kl}^{ij} = \left( P_k^i P_l^j - \frac{1}{2} P_{kl} P^{ij} \right). \quad (1.34)$$

Therefore, we can apply  $\Lambda_{kl}^{ij}$  to Equation ?? to get the transverse-traceless perturbation. For a gravitational wave travelling in the z-direction with  $n^i = (0, 0, 1)$  (referred to as the "radiation" frame) we get plus and cross polarisations as functions of  $\ddot{M}_{ij}$

$$h_+^{TT} = \frac{G}{r} (\ddot{M}_{xx} - \ddot{M}_{yy}) \quad (1.35)$$

---



$$h_{\times}^{TT} = \frac{2G}{r} (\ddot{M}_{xy}) . \quad (1.36)$$

The total power radiated by a source is the *luminosity flux*, and it must equal the energy carried by the gravitational waves, due to the conservation of energy. We can obtain the luminosity flux by integrating the energy flux over a sphere with infinite radius from the source. The total energy loss rate is expressed as

$$\frac{dE}{dt} = -\mathcal{L}, \quad (1.37)$$

where  $\mathcal{L}$  is the gravitational-wave luminosity flux. The luminosity flux at a large distance from the source is given by [? ]

$$L = \frac{1}{5} \left\langle \ddot{\dot{M}}^{jk} \ddot{M}_{jk} \right\rangle, \quad (1.38)$$

where the angular brackets represent averaging over all directions, and  $\ddot{M}_{jk}$  refers to the third time derivative of the quadrupole moment. This shows that any source with a non-vanishing second derivative of its mass quadrupole moment will emit gravitational waves.

Thus, through the quadrupole formula and gravitational-wave luminosity, we see that rapidly accelerating masses with a non-zero quadrupole moment are ideal sources of gravitational radiation.

## 1.2 Gravitational-wave detection

Having described the effect of gravitational waves on a ring of particles, we now describe how these principles are applied to detect gravitational waves on Earth. The first direct detection of gravitational waves was achieved on the 14th September 2015 [? ], made possible by ground-based gravitational-wave observatories. The current network of gravitational-wave detectors includes the Advanced LIGO (aLIGO) detectors located in Hanford and Livingston [? ]; Virgo in Italy [? ]; KAGRA in Japan [? ]; and GEO600 in Germany [? ]. While this section will primarily focus on the design and operation of the aLIGO interferometers, the underlying detection principles are similar across all observatories. The configuration and performance of aLIGO are described comprehensively in [? ].



### 1.2.1 Laser interferometry

The physical effect of a passing gravitational wave on a ring of particles (Figure ??) is described as a stretching and squeezing, changing the particles' proper distances depending on the polarisation of the gravitational wave. The gravitational-wave observatories are Michelson interferometers, and we can visualise the L-shape of them as experiencing similar effects to the ring of particles. A basic diagram of the LIGO interferometer design can be seen in Figure ??.

Half the light from the laser incident on the beam splitter is transmitted to the end test mass in the  $y$ -direction ( $\text{ETM}_N$ ) and half to the end test mass in the  $x$ -direction ( $\text{ETM}_E$ ). The end test masses are mirrors which reflect the light back toward the beam splitter. When no gravitational wave is present, the length of the  $x$  and  $y$  arm cavities are equal and the light returning to the beam splitter interferes destructively, meaning no light is detected by the photo-detector output of the detector.

Using Equation ??, where we are treating the detector arms like two particles placed at  $(L, 0, 0)$  for the end test mass of the  $x$  arm and  $(0, L, 0)$  for the  $y$ -arm, we can calculate the change in arm length (proper distance) with respect to the beam splitter at the origin under the influence of a gravitational wave propagating in the  $z$ -direction

$$ds_x \approx \left(1 + \frac{1}{2}h_+\right)L, \quad (1.39)$$

where  $L$  is the length of the detector arm (4 km for advanced LIGO) and in the  $y$  arm

$$ds_y \approx \left(1 - \frac{1}{2}h_+\right)L. \quad (1.40)$$

The strain experienced by the  $x$  and  $y$  arms is directly related to both the magnitude of the  $h_+$  gravitational-wave polarisation and the length of the detector arm. The arm in the  $x$ -direction will experience stretching, while the arm in the  $y$ -direction will experience squeezing.

The coupling of motion in both arms leads us to require the use of the differential change in arm length to calculate the gravitational-wave strain,

$$h(t) = \frac{\delta L_1 - \delta L_2}{L}, \quad (1.41)$$

where  $\delta L_1$  and  $\delta L_2$  are the changes in the lengths of the arms.

We generalise  $h(t)$  for any gravitational wave of any polarisation. The strain induced in a single arm pointing in the  $\hat{\mathbf{n}}$  direction, where  $\hat{\mathbf{n}}$  is a unit normal vector,

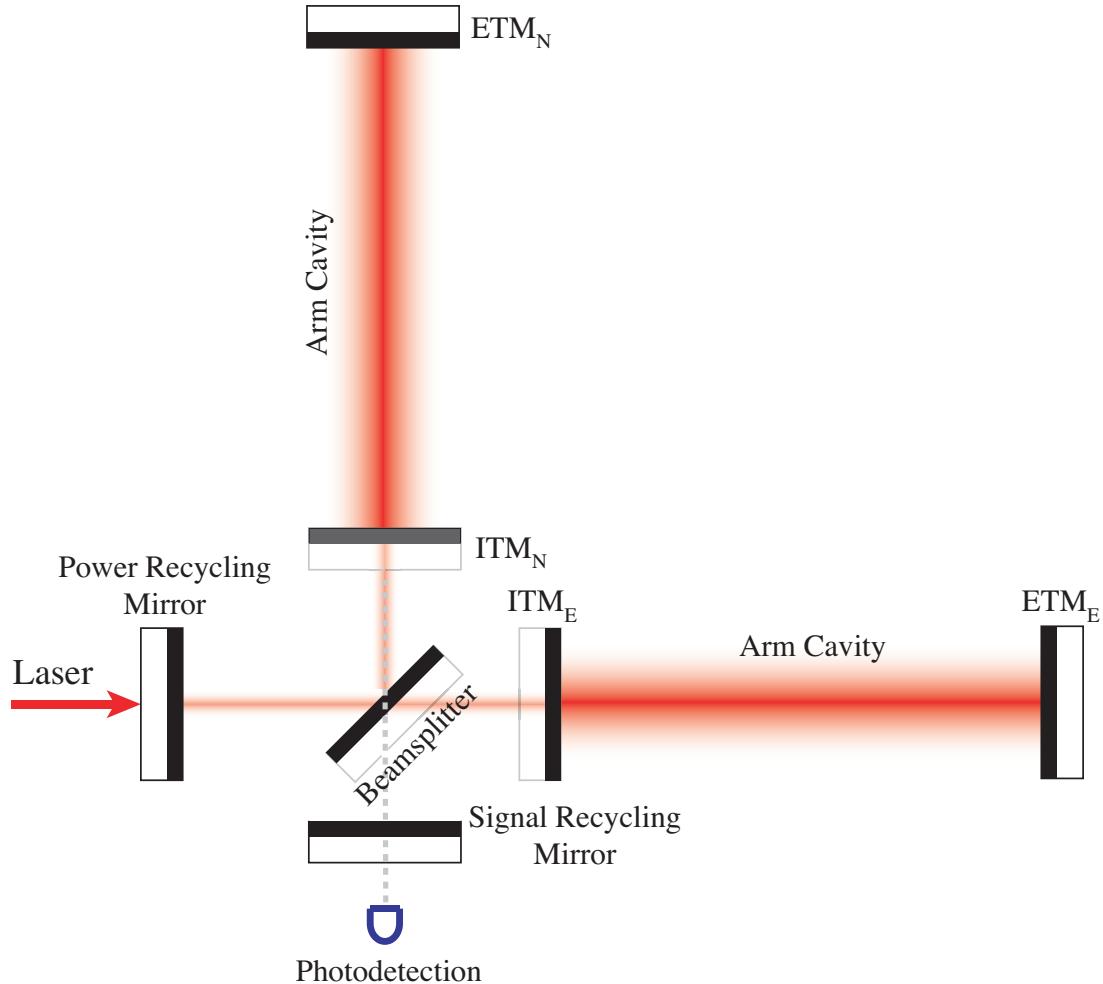


Figure 1.2: An example Michelson laser interferometer detailing the main components of the advanced LIGO gravitational-wave observatory [? ]. Half the laser is reflected to the north arm cavity, physically defined by the input test mass ( $ITM_N$ ) and the end test mass ( $ETM_N$ ), and the other half is transmitted through to the east arm. Laser power builds up in the arm cavities through reflection between test masses before returning to the beam splitter, where there will be a total destructive interference of light when no gravitational-wave signal is present and therefore no signal will be measured at the photo-detector output. The power recycling mirror reflects laser light back into the detector arms to build up more power. The signal recycling mirror reflects light back into the arms with frequencies like those expected from gravitational-wave signals which resonates inside the cavity and increase the signal sensitive frequency power. Taken from [? ].



is

$$\frac{\delta L}{L} = \frac{1}{2} h_{ij} n^i n^j \quad (1.42)$$

and the difference in lengths for two interferometer arms is

$$h(t) = \frac{1}{2} h_{ij} n_1^i n_1^j - \frac{1}{2} h_{ij} n_2^i n_2^j, \quad (1.43)$$

where crucially  $h_{ij}$  is measured in the detector's reference frame. If our detector is located in the  $x - y$  plane, the gravitational-wave signal is entirely polarised in the  $h_+$  direction.

The gravitational wave signal observed will depend on the orientation between source and detector. To transform between radiation frame and detector frame, we perform a set of rotations by angles  $(\theta, \phi, \psi)$  to adjust for the orientation and polarisation of the wave. Figure ?? shows the required angles to describe effects on the gravitational-wave signal due to these transformations. Suppose we have a gravitational-wave signal in the radiation frame, to transform into the detector frame we must first define a rotation tensor

$$R_j^i = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (1.44)$$

such that when transforming the spatial components of  $h_{ij}$  using this tensor we can evaluate the detector strain as a function of the  $+$  and  $\times$  polarisations in the radiation frame and the three angles relating the radiation frame to the detector frame. A detector with arms aligned in the  $x$  and  $y$  directions will give

$$h(t) = F_+(\Theta, \Phi, \Psi) h_+(t) + F_\times(\Theta, \Phi, \Psi) h_\times(t), \quad (1.45)$$

where,

$$F_+((\Theta, \Phi, \Psi)) = \frac{1}{2}(1 + \cos^2(\Theta)) \cos 2\Phi \cos 2\Psi - \cos \Theta \sin 2\Phi \sin 2\Psi, \quad (1.46)$$

$$F_\times((\Theta, \Phi, \Psi)) = -\frac{1}{2}(1 + \cos^2(\Theta)) \cos 2\Phi \cos 2\Psi - \cos \Theta \sin 2\Phi \sin 2\Psi \quad (1.47)$$

which are the antenna responses of the detector to the  $+$  and  $\times$  polarisations respectively, which describe the sensitivity of the detector to gravitational waves at different sky positions.

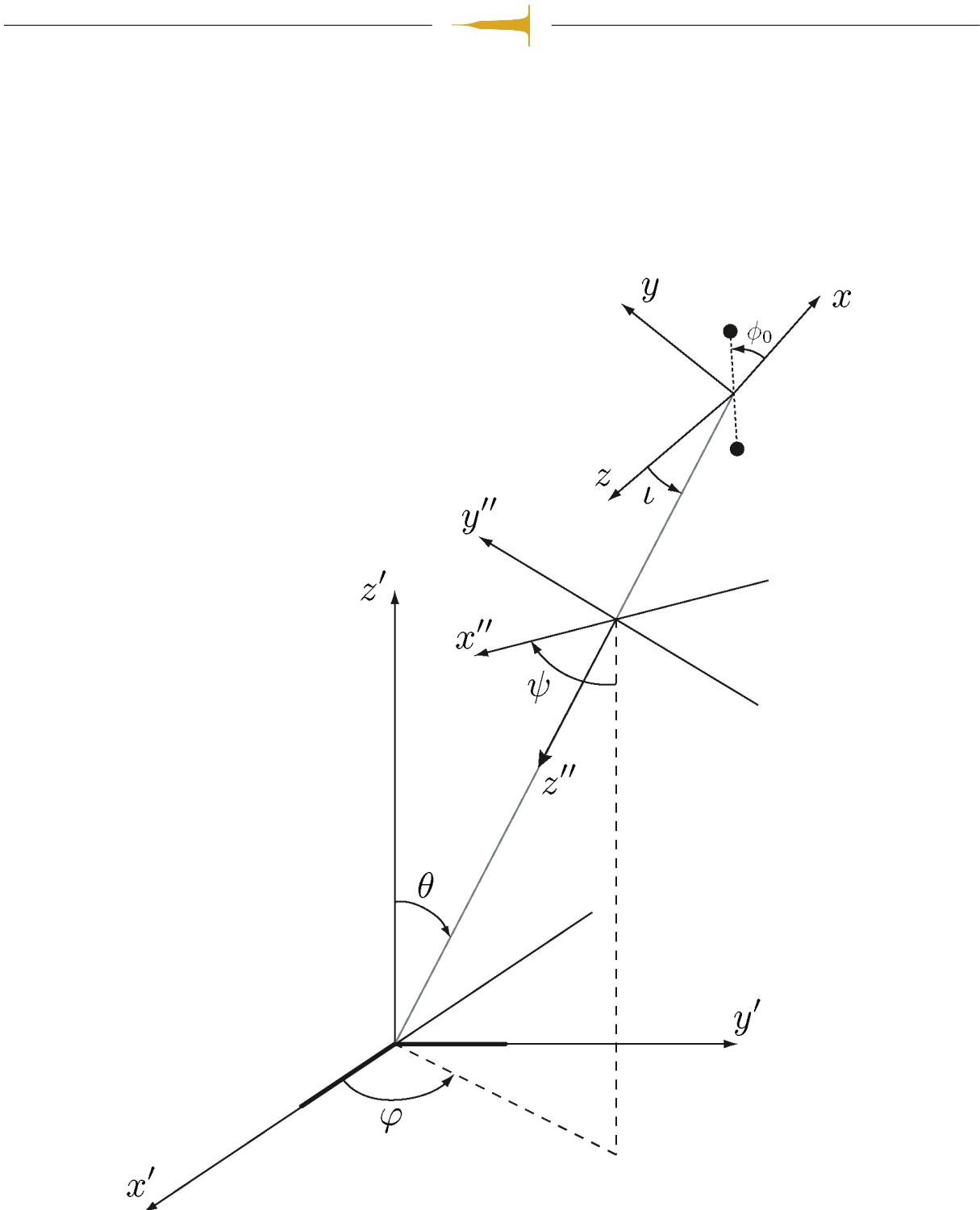


Figure 1.3: The angles of rotation to describe a gravitational wave from the source from (unprimed coordinates) to the detector frame (single primed coordinates) through the radiation frame (double primed coordinates). Taken from [? ].



### 1.2.2 Increasing detector sensitivity

With detector arm lengths on the order of  $\sim 10^3$  m and a laser wavelength of approximately  $1\text{ }\mu\text{m}$ , we are only able to measure a fractional change in the arm length on the order of  $10^{-9}$  m. However, our detectors are capable of measuring induced strain from passing gravitational waves as small as  $10^{-22}$ . In this section, we discuss the detector improvements that enable these detections.

One technique used to effectively increase the arm length of the detectors is to allow the light to travel down the arms multiple times. The Fabry-Pérot cavities [? ] in the arms of the aLIGO detectors include an additional mirror on the input test mass (ITM), facing the end mirrors, which is highly reflective, allowing only a small fraction of light to transmit through. As a result, the light reflects multiple times between these two mirrors, increasing the optical path length to approximately 1000 km. This also has the effect of building up laser power in the arms, sharpening the interference fringes to improve the ability to distinguish between noise and signal at the photodetector [? ].

Two additional components that increase detector sensitivity are: *power recycling*, where a mirror at the laser input port redirects reflected light back into the interferometer, boosting the circulating power within the arms and enhancing the signal-to-noise ratio, enabling the detection of weaker gravitational-wave signals. The second improvement is the *signal recycling mirror*, placed at the output port, which resonates with specific gravitational-wave frequencies, effectively increasing the detector's sensitivity to signals of interest. By tuning this mirror, the interferometer's sensitivity can be optimized for certain frequency bands. These improvements together allow the detector to reach sensitivities capable of observing strain on the order of  $10^{-22}$  for short-duration signals up to a few seconds long.

## 1.3 Modelling compact binary coalescences

The primary sources of gravitational waves seen by Earth-based gravitational-wave detectors are those from compact binary coalescences. Compact binary coalescences (CBC) are the inspiral and merger of two compact objects (black holes or neutron stars). The binary system will emit gravitational waves and slowly dissipate the orbital energy of the system. The system is locked in the cycle of a decreasing orbit leading to greater acceleration, increasing the rate at which energy is being dissipated and the orbital radius decreases. The orbit will decay until an eventual merger between the two objects. We have observed three distinct CBC systems: binary black holes [? ]



(BBH), binary neutron stars [? ] (BNS) and neutron star — black hole binaries [? ].

In this section, we describe the gravitational waves emitted by a CBC and the physical effects these would have on gravitational-wave detectors. First, we need to describe the intrinsic parameters of the gravitational-wave signal being created and the extrinsic, observational parameters which affect the signal based on how it is being observed. Next, we derive a gravitational waveform with a very simple binary system of two point-like masses using the Newtonian formalism.

### 1.3.1 Waveform parameters

#### Intrinsic parameters

The intrinsic parameters are those which have physical effects on the system and alter how the gravitational-wave signal is created,

- Primary and secondary component masses,  $m_1$  and  $m_2$
- Primary and secondary three-dimensional spin vectors,  $\vec{s}_1$  and  $\vec{s}_2$

Here, the subscript  $_1$  refers to the more massive object in the binary system, and  $_2$  to the less massive one.

#### Extrinsic parameters

The extrinsic parameters affect how the gravitational-wave signal is observed,

- Right ascension of the source location,  $\alpha$
- Declination of the source location,  $\delta$
- Luminosity distance to the source,  $r$ ,
- Inclination angle between the line of sight and the orbital angular momentum of the binary,  $\iota$
- Polarisation angle of the gravitational wave,  $\psi$
- Time of coalescence,  $t_c$
- Coalescence phase,  $\phi_c$

There are additional parameters that describe other physical effects such as: tidal deformations, eccentricity, the neutron star equation of state, or potential deviations from General Relativity. In this work, we will ignore these additional effects and focus on the 15 parameters listed above.



### 1.3.2 Simple CBC inspiral

We can derive  $h_+$  and  $h_\times$  for a binary system modelled as a decaying quasi-circular orbit of two point masses around a shared centre of mass. These two point masses are in the ‘inspiral’ regime, where they are radiating gravitational waves, travelling with speeds much less than  $c$  and, are not close to merging. We choose a frame of reference where the plane of the binary is in the  $x$ - $y$  direction and the binary is situated at a distance  $r$  from the observer.

Assuming a circular orbit, we obtain the orbital velocity in Kepler’s laws of planetary motion by equating the Newtonian gravitational force and the orbital centripetal force

$$\frac{m_1 m_2}{a^2} = \frac{\mu v^2}{a}, \quad (1.48)$$

$$v = \sqrt{\frac{m_1 + m_2}{a}}, \quad (1.49)$$

where  $m_1$  and  $m_2$  are the masses of the objects,  $\mu$  is the reduced mass of the system ( $\mu = \frac{m_1 m_2}{(m_1 + m_2)}$ ) and  $a$  is the distance between the two objects. Using  $\omega = v/a$ , we can obtain the orbital frequency

$$\omega_{\text{orbit}} = \sqrt{\frac{m_1 + m_2}{a^3}}, \quad (1.50)$$

and relate it to the rate of change in the orbital phase

$$\omega_{\text{orbit}} = \frac{d\Phi_{\text{orbit}}}{dt}. \quad (1.51)$$

Using the quadrupole formula (Equation ??), we can derive the metric perturbations caused by this system

$$h_{\mu\nu} = \frac{4}{r} \mu a^2 \omega_{\text{orbit}}^2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\cos(2\omega_{\text{orbit}}t) & -\sin(2\omega_{\text{orbit}}t) & 0 \\ 0 & -\sin(2\omega_{\text{orbit}}t) & \cos(2\omega_{\text{orbit}}t) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (1.52)$$

The line of sight between the binary system and the observer is defined by the inclination angle,  $\iota$ , and when projecting the gravitational-wave emission in the line of sight we get the waveform polarisations,

$$h_+(t) = \frac{4}{r} M \eta v^2 \left( \frac{1 + \cos^2 \iota}{2} \right) \cos(2\omega_{\text{orbit}}t + 2\Phi_c), \quad (1.53)$$



$$h_{\times}(t) = \frac{4}{r} M \eta v^2 \left( \frac{\cos \iota}{2} \right) \sin (2\omega_{orbit} t + 2\Phi_c), \quad (1.54)$$

where  $M = m_1 + m_2$  is the total mass of the system,  $\eta = \frac{m_1 m_2}{(m_1 + m_2)^2}$  is the symmetric mass ratio and  $\Phi_c$  is the coalescence phase of the system. A key observation from these equations is that the frequency of the gravitational waves is twice that of the binary orbital frequency.

We have an embedded assumption that the separation distance of the two objects will remain constant. We know that the radiation of gravitational waves will decrease the orbit of the binary system, and so by solving equations ?? and ??, we can find the power radiated by the binary system. The energy loss will increase the binary orbital frequency, from this we can calculate the change in phase and amplitude as a function of time.

In the two point source example we relate the orbital velocity,  $\omega_{orbit}$  to the particle separation,  $a$ , using Equation ?? and then when introducing the *chirp mass*,  $\mathcal{M}$ , of the binary system

$$\mathcal{M} = \frac{(m_1 m_2)^{\frac{3}{5}}}{(m_1 + m_2)^{\frac{1}{5}}}, \quad (1.55)$$

and solving equations ?? and ?? we are able to determine the power radiated by this system as

$$\frac{dE}{dt} = -\frac{32}{5} (\mathcal{M} \omega_{orbit})^{\frac{10}{3}}. \quad (1.56)$$

The total energy of the system can be expressed as

$$E = -\frac{m_1 m_2}{2a}, \quad (1.57)$$

$$= -\frac{1}{2} (\mathcal{M}^5 \omega_{orbit}^2)^{\frac{1}{8}}, \quad (1.58)$$

and when substituted into Equation ?? and solving we obtain the orbital frequency as a function of time

$$\omega_{orbit}(t) = \left( -\frac{256}{5} \mathcal{M}^{\frac{5}{3}} (t - t_0) \right)^{-\frac{3}{8}}, \quad (1.59)$$

$$= \frac{1}{8} \left( \frac{\tau}{5} \right)^{-\frac{3}{8}} \mathcal{M}^{-\frac{5}{8}}, \quad (1.60)$$

where  $t_0$  is the fiducial starting time for integration which is used to map the time to define a new time variable  $\tau = t_c - t$ .

We can integrate Equation ?? over time to find the orbital phase of the gravitational-



wave signal,

$$\Phi(\tau) = \int_{t+0}^t \omega_{orbit} dt, \quad (1.61)$$

$$= -2 \left( \frac{5G\mathcal{M}}{c^3} \right)^{-\frac{5}{8}} \tau^{\frac{5}{8}} + 2\Phi_c, \quad (1.62)$$

where  $\Phi_c$  is the coalescence phase.

Finally, combining the previous equations, we have derived

$$h_+(\tau) = \frac{1}{r} \left( \frac{G\mathcal{M}}{c^2} \right)^{\frac{5}{4}} \left( \frac{5}{c\tau} \right)^{\frac{1}{4}} \left( \frac{1 + \cos^2 \iota}{2} \right) \cos \Phi(\tau), \quad (1.63)$$

$$h_\times(\tau) = \frac{1}{r} \left( \frac{G\mathcal{M}}{c^2} \right)^{\frac{5}{4}} \left( \frac{5}{c\tau} \right)^{\frac{1}{4}} \cos \iota \sin \Phi(\tau), \quad (1.64)$$

the gravitational-wave polarisations for a metric perturbation originating from our simple two-point particle binary system. From these equations, it's easy to show that both the amplitude and frequency of the system will increase rapidly and "chirp" as it approaches coalescence. Also clear to see is that these gravitational-wave signals are dependent only on the chirp mass of the system.

### 1.3.3 Fourier transform of the chirp signal

In gravitational-wave analysis, the Fourier transform of a chirp signal is crucial for understanding its frequency domain characteristics. We consider a gravitational waveform given by

$$h(t) = A(t) \cos \Phi(t), \quad (1.65)$$

where  $A(t)$  represents the amplitude, and  $\Phi(t)$  is the phase of the signal. To simplify our analysis, we use the *stationary phase approximation*, applicable when the amplitude  $A(t)$  varies slowly compared to the phase  $\Phi(t)$ . Under this approximation, the Fourier transform of  $h(t)$  is given by

$$\tilde{h}(f) \simeq \frac{1}{2} A(t(f)) \left( \frac{dt(f)}{df} \right)^{\frac{1}{2}} e^{i(2\pi f t(f) - \Phi(f) - \frac{\pi}{4})}, \quad (1.66)$$

where  $t(f)$  and  $\Phi(f)$  are the time and phase at frequency  $f$ . For a chirp signal, these quantities can be expressed as

$$t(f) = t_c - \frac{5}{8\pi} (8\pi f)^{-\frac{8}{3}} \mathcal{M}^{-\frac{5}{3}}, \quad (1.67)$$



$$\Phi(f) = \Phi_c - 2(8\pi\mathcal{M}f)^{-\frac{5}{3}}, \quad (1.68)$$

where  $t_c$  is the coalescence time,  $\mathcal{M}$  is the chirp mass, and  $\Phi_c$  is the phase at coalescence.

The Fourier transform of the chirp signal, considering the Newtonian approximation, can be expressed in the form of

$$\tilde{h}(f) = \sqrt{\frac{5}{24}} \frac{G^2 \mathcal{M}^2}{c^5 d_L} \left( \frac{\pi G \mathcal{M} f}{c^3} \right)^{-\frac{7}{6}} \left( \frac{1 + \cos^2 \iota}{2} \right) e^{i\Phi(f; t_c)}, \quad (1.69)$$

where the phase term  $\Phi(f; t_c)$  is defined by

$$\Phi(f; t_c) = \frac{\pi}{4} - \frac{3}{128} \left( \frac{\pi G \mathcal{M} f}{c^3} \right)^{-\frac{5}{3}} - 2\pi f t_c + \Phi_c. \quad (1.70)$$

Here,  $d_L$  denotes the luminosity distance to the source, and  $\iota$  is the inclination angle of the orbital plane with respect to the line of sight.

### 1.3.4 Post-Newtonian approximation

Our Newtonian model of the gravitational-wave signal from a CBC source has been derived with simplifications. However, the real signals detected by gravitational-wave detectors are far more complex, requiring corrections to account for the effects of General Relativity. The post-Newtonian (PN) expansion technique improves the accuracy of gravitational waveforms by introducing relativistic corrections to the waveform, accounting for deviations from Newtonian gravity due to General Relativity [? ? ? ].

The PN expansion can be performed by expanding in powers of  $v/c$ , where  $v$  is the velocity of the orbiting bodies, to obtain the corrections to the waveform which can then be summed to get waveforms up to the required PN order. For a CBC, the waveform could be written as

$$h(t) = h_{\text{Newtonian}}(t) + h_{1\text{PN}}(t) + h_{1.5\text{PN}}(t) + h_{2\text{PN}}(t) + \dots, \quad (1.71)$$

where  $h_{\text{Newtonian}}(t)$  is the leading-order Newtonian term, and the following terms represent higher and higher PN order corrections [? ].

### 1.3.5 Full waveform

The post-Newtonian (PN) formalism is accurate during the inspiral phase of the waveform when the two objects are still distant from each other, and formalism does not

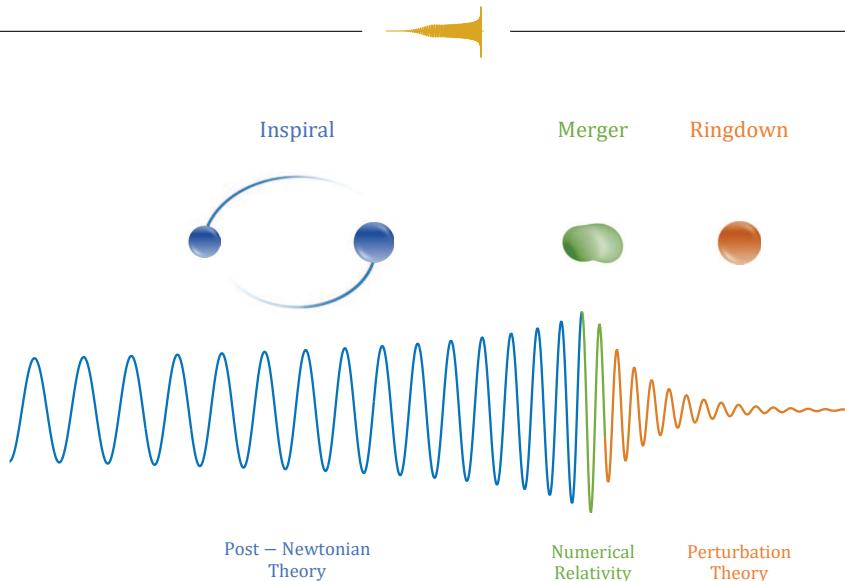


Figure 1.4: The evolution of a compact binary merger is illustrated through three distinct phases. The inspiral phase involves the two compact objects orbiting each other, with the orbit decaying as they emit gravitational waves. This phase is modelled using post-Newtonian theory. The merger phase occurs when the orbital radius decreases sufficiently for the objects to merge into a new compact object, modelled using numerical relativity. The ringdown phase represents the newly formed object’s vibration as it settles into its final shape, modelled using perturbation theory. This image is adapted from [? ].

cover the entire evolution of the system. The system continues to emit gravitational waves throughout the merger and post-merger phases, including the ringdown phase, where the newly formed object settles into its final shape. The gravitational-wave emission peaks during the merger, making it crucial for detecting gravitational waves. The ringdown phase is particularly valuable for testing General Relativity (GR) [? ? ? ].

Generating full waveforms requires solving the Einstein field equations using numerical relativity (NR), a process that can take weeks. To efficiently model the entire evolution of the waveform—comprising the inspiral, merger, and ringdown (IMR) phases (see Figure ??)—waveform families use a combination of NR waveforms and analytical or semi-analytical approximations. This approach balances accuracy with computational efficiency. The following subsections briefly describe these waveform models.

### Effective-One-Body (EOB)

The Effective-One-Body (EOB) model [? ] simplifies the binary system by transforming it into an equivalent single test particle moving in an external gravitational field, described by a Hamiltonian [? ? ? ? ]. This Hamiltonian includes both leading-order



Newtonian terms and higher-order post-Newtonian corrections [? ].

EOB models tuned to numerical relativity (NR) data are known as “EOBNR” models [? ]. Some of these models, such as SEOBNRv5 [? ], also incorporate the effects of anti-aligned spins. Further enhancements allow for precession effects and higher-order modes of gravitational-wave emission, as seen in the state-of-the-art model SEOBNRv5\_PHM [? ], which represents the most advanced EOB waveform model as of the drafting of this thesis.

### Phenomenological models

Phenomenological models, often abbreviated as ‘Phenom’, use a phenomenological ansatz calibrated with NR simulations to produce full waveforms. These models are fully analytical and can generate both Fourier domain and time domain waveforms [? ? ]. The ability to create waveforms directly in the Fourier domain is highly desirable for analyses that require generating millions of waveforms, such as offline searches or parameter estimation.

Phenom models that cover the inspiral, merger, and ringdown phases are prefixed with `IMRPhenom` [? ]. The state-of-the-art model of this family is `IMRPhenomXPHM` [? ], which can accurately model anti-aligned spins, precessing binaries (P), higher-order modes (HM), and extreme mass ratios (X).

### Surrogate models

Surrogate models create waveforms by interpolating from NR simulations [? ? ]. These models, such as `NRSur7dq4` [? ], provide greater accuracy than other model families but are limited by the parameter regions covered by the NR simulations used for interpolation. The current leading surrogate model, `NRSur7dq4`, is valid up to a mass ratio of four.



# 2

---

## Searching for Gravitational Waves from Compact Binary Coalescences

---

*“Even in death I serve the Omnissiah”*

— Magos Dominus Reditus, Warhammer 40,000: Mechanicus

Gravitational-wave search pipelines, such as PyCBC, play a critical role in analysing data from detectors to identify gravitational-wave signals, both in real time (live) and in post-processing (offline). This chapter develops the theoretical foundation for these search pipelines and explores the techniques employed in gravitational-wave detection. Emphasis is placed on the PyCBC pipelines—PyCBC Offline and PyCBC Live—both of which were extensively utilised and further developed as part of this thesis.

### 2.1 Gravitational-wave data

Gravitational-wave observatories produce a dimensionless strain time-series,  $s(t)$ , which is composed of detector noise,  $n(t)$ , and, when present, an astrophysical gravitational-wave signal,  $h(t)$

$$s(t) = \begin{cases} n(t), & \text{if no signal is present,} \\ n(t) + h(t), & \text{if a signal is present.} \end{cases} \quad (2.1)$$



The primary objective of gravitational-wave search pipelines is to extract  $h(t)$  from  $s(t)$ , detecting the astrophysical signal from within the background noise.

## 2.2 Searching through gravitational-wave data

Detecting gravitational waves from compact binary coalescences requires sophisticated search methods capable of sifting through vast amounts of data collected by gravitational-wave detectors. This section details the search techniques used to identify and characterise gravitational-wave signals. There are many pipelines that search for gravitational waves [? ? ? ? ? ? ? ? ] and in this section we will focus on the PyCBC search [? ? ? ]

We begin with the single-detector search techniques to identify potential gravitational-wave events, and then detail the tests and methods used to combine single detector detections to identify coincidentally found gravitational-wave signals.

### 2.2.1 Matched filtering

Detecting gravitational waves relies on being able to distinguish between data that contains only noise,  $s(t) = n(t)$ , and data which contains a gravitational-wave signal,  $s(t) = n(t) + h(t)$ . We must construct an optimal detection statistic, which expresses the value of the probability that the data contains a known signal. The Neyman-Pearson likelihood ratio forms the optimal detection statistic [? ] and is found by computing the ratio of the probability that the data contains the signal (hypothesis  $\mathcal{H}_1$ ) to the probability that the data is pure noise (the null hypothesis  $\mathcal{H}_0$ )

$$\Lambda = \frac{P(s|\mathcal{H}_1)}{P(s|\mathcal{H}_0)}, \quad (2.2)$$

where  $\Lambda$  is the likelihood ratio that serves as the detection statistic,  $P(s|\mathcal{H}_1)$  is the probability that the data contains the signal, and  $P(s|\mathcal{H}_0)$  is the probability that the data is pure noise. It is natural to use probability densities due to the detection process involving continuous data and not discrete events

$$\Lambda = \frac{p(s|\mathcal{H}_1)}{p(s|\mathcal{H}_0)}, \quad (2.3)$$

where  $p(s|\mathcal{H}_1)$  is the probability density that the data contains the signal, and  $p(s|\mathcal{H}_0)$  is the probability density that the data is pure noise.

We first define the noise-weighted inner product between two time series  $s(t)$



(data) and  $h(t)$  (signal) as [? ]

$$(s|h) = 4\Re \int_0^\infty \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_n(f)} df, \quad (2.4)$$

where a tilde denotes a Fourier transformed version of the variable and where  $S_n(f)$ , represents the one-sided power spectral density (PSD) of the data, defined as

$$\langle \tilde{s}(f)\tilde{s}(f') \rangle = \frac{1}{2}S_n(f)\delta(f-f'), \quad (2.5)$$

where angle brackets denote an average over noise realisations and  $\delta$  is the Dirac delta function. It is important to note,  $p(a|b)$  denotes conditional probability and  $(a|b)$  denotes the noise-weighted inner product.

If the detector noise is Gaussian, we can compute the probability densities

$$p(s|\mathcal{H}_0) \propto e^{-(s|s)/2}, \quad (2.6)$$

$$p(s|\mathcal{H}_1) \propto e^{-(s-h|s-h)/2}, \quad (2.7)$$

and the likelihood ratio (Equation ??)

$$\Lambda(\mathcal{H}_1|s) = \frac{e^{-(s-h|s-h)/2}}{e^{-(s|s)/2}} = e^{(s|h)}e^{-(h|h)/2}. \quad (2.8)$$

The exponential scaling of the likelihood values can introduce numerical instability when computing the likelihood ratio. Therefore, we take the logarithm, resulting in the log-likelihood ratio

$$\log \Lambda(\mathcal{H}_1|s) = (s|h) - \frac{(h|h)}{2}. \quad (2.9)$$

From Equation ?? we can see that the likelihood ratio depends only on the data through the inner product of  $s$  and  $h$  and is the optimal detection statistic known as the *matched filter*, effectively a noise-weighted correlation between the known signal and the data.

Since we are interested in evaluating the presence of a signal in the data, it is useful to define the *signal-to-noise ratio* (SNR),  $\rho$ , which quantifies how strong the signal is relative to the background noise. The SNR of a known signal in data can be derived as [? ]

$$\rho = \frac{(s|h)}{\sqrt{(h|h)}}. \quad (2.10)$$

This expression indicates how well the signal correlates with the data relative to the noise level in the detector. When the detector contains a real signal, a high SNR



corresponds to a stronger, more detectable signal, whereas a low SNR indicates a weak signal buried in the noise. We will use the SNR value as the detection statistic moving forward.

## 2.2.2 Signal-to-noise ratio over time

We have assumed that we know all the parameters of the signal we are searching for; however, this is not likely for real gravitational-wave signals. We will begin to build up a more robust search, in which we know very few of the initial gravitational-wave signal parameters. First, we consider the case of a signal that has a known waveform but an unknown amplitude and arrival time. We can describe the true signal as

$$h(t) = Ag(t - t_0), \quad (2.11)$$

where  $A$  is the unknown amplitude of the signal,  $t_0$  is the unknown arrival time and  $g(t)$  is the known waveform. We can take the Fourier transform of this signal

$$\tilde{h}(f) = A\tilde{g}(f)e^{-2\pi ift_0}, \quad (2.12)$$

and obtain the matched filter using Equation ?? to be

$$(s|h) = 4A\Re \int_0^\infty \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_n(f)} e^{2\pi ift_0} df. \quad (2.13)$$

From this we can define

$$(s|h) = Ax(t_0) \quad \text{where,} \quad (2.14)$$

$$x(t) = 4\Re \int_0^\infty \frac{\tilde{s}(f)\tilde{g}^*(f)}{S_n(f)} e^{2\pi ift} df. \quad (2.15)$$

$x(t)$  is a time series representing the matched filter at any arrival time  $t$ . From this we can define an SNR time series  $\rho(t)$  containing information about the SNR value at each point in time. To find the maximum likelihood detection statistic we simply find the largest value of  $\rho(t)$  which will correspond to the amplitude and will reveal the previously unknown arrival time  $t_0$ .



### 2.2.3 Phase maximisation

The phase of the gravitational-wave signal is another unknown parameter. In Section ?? we give a signal of the form

$$h(t) = A(t) \cos(\Phi(t)), \quad (2.16)$$

and can include an additional phase offset,  $\phi_0$ , to account for the random orientation and sky position of the binary

$$h(t) = A(t) \cos(\Phi(t) + \phi_0). \quad (2.17)$$

We can maximise over this phase offset using the matched filter by rewriting our signal as

$$h(t) = h_0(t) \cos(\phi_0) + h_{\pi/2}(t) \sin(\phi_0), \quad (2.18)$$

where  $h_0(t)$  and  $h_{\pi/2}(t)$  are realisations of Equation ?? where the phase offset has been set equal to 0 and  $\frac{\pi}{2}$  respectively [? ].

We can then calculate  $\rho^2$  using these two new signals to maximise over the phase

$$\max_{\Phi}(\rho^2(t)) = \frac{(s|h_0)^2 + (s|h_{\pi/2})^2}{(h_0|h_0)}, \quad (2.19)$$

having made the assumption that  $\tilde{h}_{\pi/2}(f) = i\tilde{h}_0(f)$  which is true for frequency domain waveforms with the stationary phase approximation [? ]. Conventionally we take the square root of Equation ?? to obtain the SNR, where phase has been maximised over

$$\rho = \sqrt{\max_{\Phi}(\rho^2(t))}. \quad (2.20)$$

### 2.2.4 Template bank

We have demonstrated that the matched filter can be used to analytically and efficiently maximise over the amplitude, time of arrival and phase of a known signal. We acknowledge that for a real search for gravitational waves, we will not know the 15 parameters of the signal. The search is performed by creating many realisations of the gravitational-wave signal and searching over the data with each of them. However, we need to discuss how the realisation parameter values are chosen to ensure a sufficiently covered parameter space.



When a signal is found by a template<sup>1</sup> with parameters not equal to the true values, we will expect to see a fractional loss in the expected SNR. The closer the template parameters are to the signal parameters, the closer to the maximum SNR we will see.

We can define a signal with parameters  $\lambda$

$$h(t) = \rho u(t; \lambda), \quad (2.21)$$

where  $\rho$  is the expected SNR value for the true template. If we have a template with parameters  $u(t; \lambda + \Delta\lambda)$ , the expected SNR when matched filtering the signal with this template will be

$$\rho' = (h|u(\lambda + \Delta\lambda)) = \rho(u(\lambda)|u(\lambda + \Delta\lambda)), \quad (2.22)$$

and we can see therefore that the expected fractional loss in the expected SNR is

$$\frac{\rho - \rho'}{\rho} = 1 - (u(\lambda)|u(\lambda + \Delta\lambda)) = 1 - \mathcal{A}, \quad (2.23)$$

where we can define  $\mathcal{A}$  as the ambiguity function

$$\mathcal{A}(\lambda; \lambda + \Delta\lambda) := (u(\lambda)|u(\lambda + \Delta\lambda)), \quad (2.24)$$

which tells us how well our nearby template matches to the true signal<sup>2</sup>. If the ambiguity value is large, then we have a small fractional loss, and the template is a good match to the true signal; if the value is small, then the template is a poor description of the signal.

$\lambda$  contains both the *intrinsic* and *extrinsic* parameters of the signal, we can define the *overlap* between two templates  $h_1$  and  $h_2$  considering only the intrinsic parameters when we maximise over the extrinsic parameters,

$$\mathcal{O}(h_1, h_2) := (h_1|h_2) = \frac{(h_1|h_2)}{\sqrt{(h_1|h_1)(h_2|h_2)}}. \quad (2.25)$$

The overlap for a signal  $h_1$  represents the fraction of the SNR recovered by matched filtering with the template  $h_2$ . The match is then the overlap maximised over time of arrival and phase [?]

$$\mathcal{M}(h_1, h_2) = \max_{\phi_0, t_c} (\hat{h}_1|\hat{h}_2(\phi_c, t_c)). \quad (2.26)$$

We then require that any signal can be recovered with a maximum 3% [? ] SNR loss,

---

<sup>1</sup>A realisation of the gravitational-wave signal waveform.

<sup>2</sup>Templates are assumed to be normalised such that  $(u(\lambda)|u(\lambda)) = 1$ .



corresponding to a mismatch,  $1 - \mathcal{M}$ , of 0.03, at least one template must have a maximised overlap of at least 0.97 with the true signal waveform. We can then construct a bank of templates with this requirement. Template banks can be constructed using two primary approaches: geometrically [? ? ? ? ?] or stochastically [? ? ].

In a geometric approach, a lattice is defined within the template parameter space, with each lattice point corresponding to a unique template. In the stochastic approach, randomly generated templates are placed in the bank and their match to existing templates is calculated. A new template is retained if its maximum match with any existing template is less than 0.97; it is discarded if the match is greater than or equal to 0.97.

## 2.2.5 Signal consistency tests

The matched filter is described as the optimal detection statistic in stationary Gaussian noise for searching for known signals. While we have dealt with the case of an unknown signal, we now consider the case where the detector noise is not Gaussian. Within the detector data, we have many instances of short duration bursts of excess power that are non-Gaussian, commonly called ‘glitches’ [? ].

Glitches produce large SNRs in the matched filter even when they do not share the same morphological characteristics. To combat this, we use signal consistency tests, which are able to discriminate between glitches and signal based on the distribution of the power present in the detector data.

We know the expected time and frequency evolution of a gravitational-wave signal using our waveform models. The time-frequency waveform consistency test described in [? ] (the Allen  $\chi^2$  test) divides the signal template into  $p$  sub-templates such that each sub-template contributes equally to the total SNR,

$$4 \int_0^{f_1} \frac{\tilde{s}(f)\tilde{h}_1(p)^*(f)}{Sn(f)} df = 4 \int_{f_1}^{f_2} \frac{\tilde{s}(f)\tilde{h}_2(p)}{Sn(f)} df = \dots = 4 \int_{f_{p-1}}^{\inf} \frac{\tilde{s}(f)\tilde{h}_p(f)}{Sn(f)} df, \quad (2.27)$$

where  $\tilde{s}(f)$  is the data and  $\tilde{h}_p(f)$  the sub-templates from discrete non-overlapping frequencies.

To calculate the divergence from the expected time-frequency distribution we can calculate the chi-squared value evaluated at some time,  $t_0$ , using

$$\chi^2(t_0) = \sum_{i=1}^p \left( \frac{\rho}{\sqrt{p}} - \rho_{bin,i} \right)^2, \quad (2.28)$$

where  $p$  is the number of sub-templates (bins),  $\rho$  is the SNR for the full template



matched-filtered with the data and  $\rho_{bin}$  is the SNR found when matched filtering the sub-template and the data. The  $\chi^2$  value will be small for data containing the true signal but will be large when the SNR found in some bins is different from the expected SNR, notably in the presence of a glitch with power distribution that does not match the signal.

If the data noise is Gaussian the  $\chi^2$  value will be  $\chi^2$  distributed with  $2p - 2$  degrees of freedom, therefore, we use the reduced- $\chi^2$  value which will evaluate close to 1 for the true signal template

$$\chi_r^2 = \frac{\chi^2}{2p - 2}. \quad (2.29)$$

The sine-Gaussian  $\chi^2$  test is another test used by gravitational-wave searches to identify excess power in frequencies *above* the expected frequency range of the signal. To do this, multiple sine-Gaussian functions are defined across the frequency range and matched filtered with the data to find deviations from the expected SNR at each frequency [? ]. If a gravitational-wave signal merges at a specific frequency, then we do not expect a large SNR for sine-Gaussian functions at frequencies greater than the merger frequency.

These two  $\chi^2$  tests are used to re-weigh the SNR of the matched filter, first the Allen  $\chi^2$  test is applied [? ]

$$\rho = \begin{cases} \hat{\rho}, & \text{if } \chi_r^2 \leq 1, \\ \rho [(1 + (\chi_r^2)^3)/2]^{-\frac{1}{6}}, & \text{if } \chi_r^2 > 1, \end{cases} \quad (2.30)$$

then the sine-Gaussian  $\chi^2$  test is applied

$$\hat{\rho}_{sg} = \begin{cases} \hat{\rho}, & \text{if } \chi_r^2 \leq 4, \\ \hat{\rho} (\chi_{r,sg}^2)/4 \right)^{-\frac{1}{2}}, & \text{if } \chi_r^2 > 4. \end{cases} \quad (2.31)$$

We note that the coefficients used in the re-weighing are determined empirically and can be tuned for different populations (though this has not been done).

We define a detection threshold such that a template found at a time with SNR above the threshold will be considered for consideration as a gravitational-wave signal, we call these detections ‘triggers’. The re-weighted SNR is used to rank detector triggers from single detector matched filters.

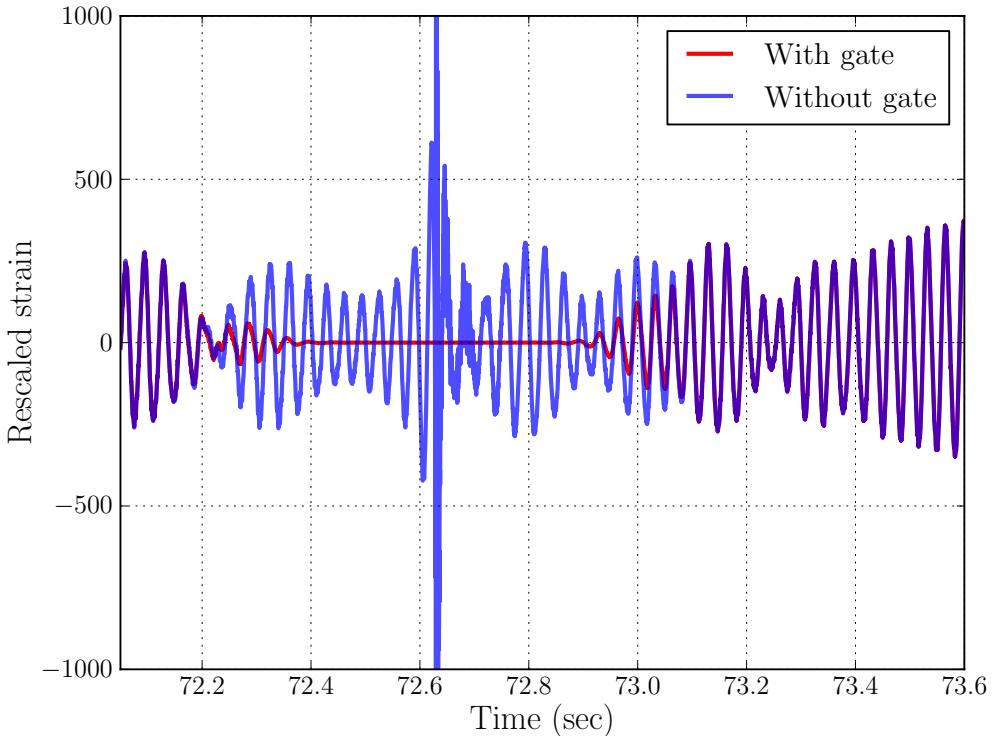


Figure 2.1: Gating a very loud noise transient. The detector strain has been rescaled by a factor of  $10^{21}$  and the glitch has a peak magnitude over 5,000. The blue line shows the data before applying the Tukey window and the red shows the data after applying the Tukey window. Note the smooth decrease in data amplitude at the edges of the windowing function. Image taken from [? ].

## 2.2.6 Auto-gating

When matched filtering the template bank and the data, glitches will produce triggers which are suppressed by the signal-consistency tests. Very loud glitches which resemble delta-functions will produce very large SNR triggers and will cause a *ringing* effect where the matched filter will continue to produce high SNR triggers for a short time.

These glitches are identified by looking for excess power in the whitened<sup>3</sup> data and are handled by applying a windowing function around the data. This process is known as *gating* and the window is applied such that a smooth transition between data and zeroing is made to avoid discontinuities in the data which can introduce more data artefacts. Figure ?? shows an example of a glitch which has been gated.

Gating occurs automatically in gravitational-wave searches in a process called ‘auto-gating’. Auto-gating in PyCBC Live zeroes only 0.25 seconds of data with a

<sup>3</sup>Whitened data has a flat power spectral density with no frequency-dependent noise.

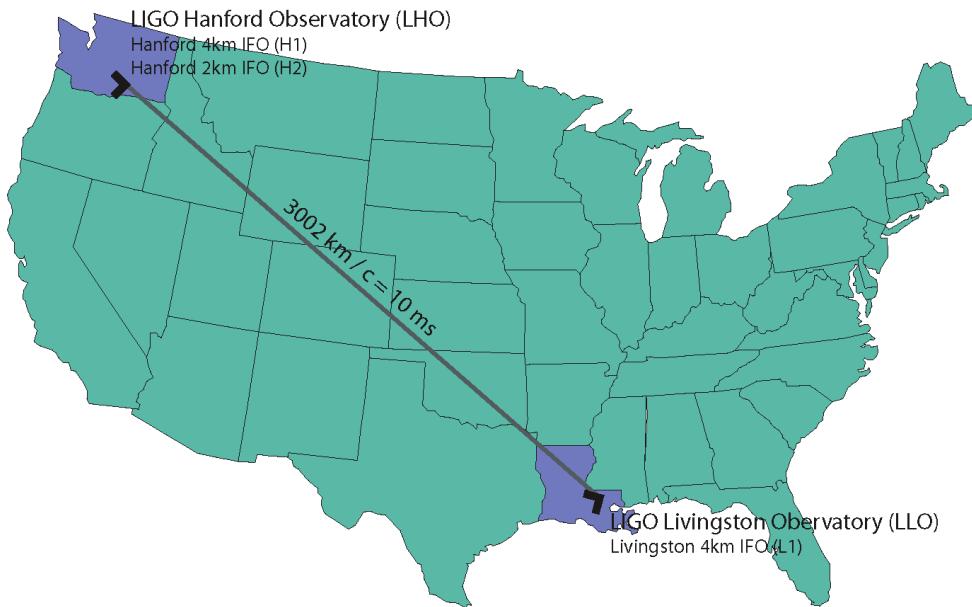


Figure 2.2: The two LIGO detectors locations and orientations, separated by approximately 3,000 km. Taken from [? ].

taper of 0.25 seconds on either side. Auto-gating will completely remove the delta-function glitch and will prevent the ringing effect, removing the triggers caused by the glitch.

### 2.2.7 Coincidence tests

Non-Gaussian transients in our detector data lead to greatly increased possibility for a gravitational-wave detector to report detections of gravitational-wave signals caused by non-astrophysical sources. If, after our signal-consistency tests, a glitch has a large SNR we are unable to make a distinction between it and a real event.

In terms of the optimal detection statistic, we must include further components which can reject glitches while ensuring we continue to detect all possible real events. The most powerful method for confirming the detection in one detector is the coincidental detection of the same signal in another detector. We call this a *coincidence test*; the requirement that for a signal to be considered real, it must have been observed in multiple detectors.

In a two detector example, for example LIGO-Hanford and LIGO-Livingston, the signal seen by both detectors will not be exactly the same. The detectors are located approximately 3,000 km from one another, (Figure ??) and we know gravitational waves propagate at the speed of light (Section ??). Therefore, we can expect a maximum delay between single-detector detections of 10 ms. For the coincidence test, we must allow a window between single detector triggers of this light travel time. An



additional coincidence requirement in PyCBC is the intrinsic parameters of the two single detector triggers being the same, i.e. they have seen the same gravitational waveform.

## 2.2.8 Ranking statistic

Finally, we can combine all these components to create a ranking statistic to use as the detection statistic for identifying gravitational-wave signals from gravitational-wave detector data. The likelihood ratio is calculated using the signal and noise event rate densities, defined as [? ]

$$r_s(\vec{\kappa}) = \frac{\text{Number of signal events}}{\text{Volume} \times \text{Time}}, \quad (2.32)$$

$$r_n(\vec{\kappa}) = \frac{\text{Number of noise events}}{\text{Volume} \times \text{Time}}. \quad (2.33)$$

We compare the number of signal and noise events over the total Volume-Time (VT) observed, the number of signal events should increase with an increase in both volume and time, but the number of noise events will not change with an increase in volume, only time. We use the volume in the noise event rate density calculation to allow us to simply compare the two rates as the VT cancels in likelihood ratio

$$\Lambda(\vec{\kappa}) = \frac{r_s(\vec{\kappa})}{r_n(\vec{\kappa})}, \quad (2.34)$$

where  $\vec{\kappa}$  is a set of parameters

$$\vec{\kappa} = \left\{ \left[ \rho_d, \chi_d^2, \chi_{d,sg}^2, \sigma_d \right], \vec{\theta}, [\mathfrak{A}_{d_1 d_2}, \delta t_{d_1 d_2}, \delta \phi_{d_1 d_2}] \right\}. \quad (2.35)$$

$\vec{\kappa}$  contains three categories of parameter: first, the single detector trigger parameters for each detector  $d$ , SNR, Allen- $\chi^2$ , sine-Gaussian  $\chi^2$  and template sensitivity ( $\sigma_d$ ); next, the intrinsic template parameters  $\vec{\theta}$  and; the coincident trigger parameters, amplitude ratio ( $\mathfrak{A}_{ab}$ ), time of arrival difference ( $\delta t_{ab}$ ) and phase difference ( $\delta \phi_{ab}$ ) between two detectors  $d_1$  and  $d_2$ .

Again, it is natural to use the log-likelihood ratio due to the order of magnitude difference between the event rate densities

$$R(\vec{\kappa}) = \log \Lambda(\vec{\kappa}) = \log r_s(\vec{\kappa}) - \log r_n(\vec{\kappa}). \quad (2.36)$$

We call the log-likelihood detection statistic the ranking statistic, which is constructed



from a noise model and signal model. The PyCBC noise and signal models have changed over the course of the gravitational-wave search history. We will describe the models used in [? ], the first iteration of the PyCBC search for compact binary coalescences.

## PyCBC

The PyCBC search pipeline ranked single detector triggers by their SNR and  $\chi^2$  values to create a re-weighted SNR,  $\hat{\rho}_d$ , for each detector. The PyCBC search pipeline ranking statistic noise event rate model provides information on how likely an event is to be caused by noise. The noise event rate model used by the PyCBC search pipeline was a simple quadrature sum of single detector re-weighted SNRs,

$$\hat{\rho}^2 = \sqrt{\hat{\rho}_{d_1}^2 + \hat{\rho}_{d_2}^2}, \quad (2.37)$$

where a large  $\hat{\rho}^2$  indicates a more significant event, less likely to be caused by noise.

The PyCBC search pipeline had no signal event rate model, and so the ranking statistic was simply the noise event rate model presented above. As detailed in [? ], this ranking statistic was shown to ‘downrank’ all triggers below a re-weighted SNR of 6 in both a Gaussian noise simulation and the 6<sup>th</sup> science run of LIGO [? ]. We note that in this case there are 0 real signals in the data and therefore this noise model has been applied to only noise triggers.

### 2.2.9 Candidate event significance

We can define some threshold at which a coincident trigger with ranking statistic value above this threshold can be preliminary considered to be a gravitational-wave event, we refer to these coincident triggers as *candidates*. We define the *significance* of the candidate as the *false-alarm rate*. False alarms are coincident triggers that have been cause entirely by noise, with no astrophysical origin. The rate of false alarms depends on the search pipeline’s response to detector noise and must be measured empirically.

The PyCBC search pipeline measures the false-alarm rate using ‘time slides’. As discussed in Section ??, a coincident trigger can only be formed if the triggers are within the light travel time window. Therefore, if a coincidence is made between two single detector triggers outside this window it **must** be caused by detector noise and not an astrophysical signal. To measure the background rate of coincident triggers, we can take the triggers from the first detector and count all the coincidences made with

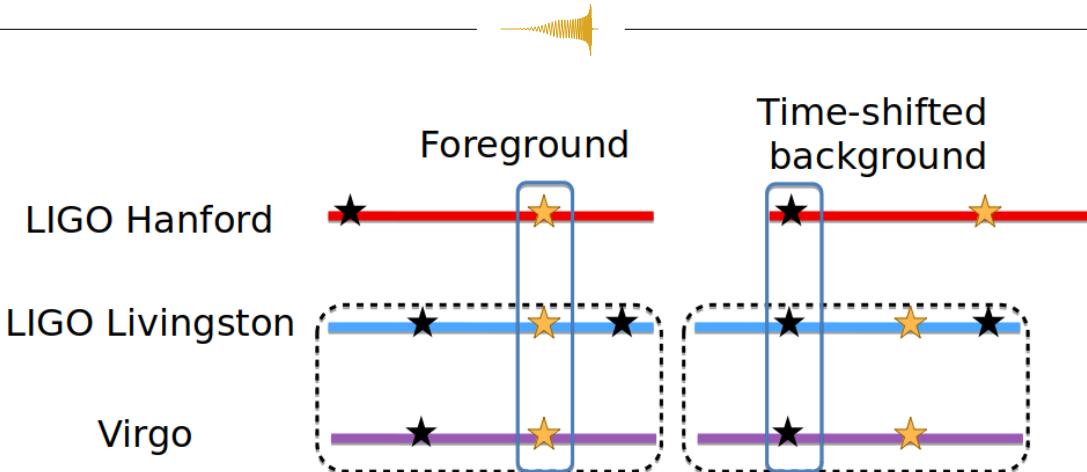


Figure 2.3: The time-sliding process carried out by the PyCBC search to generate a background distribution of candidate events. The foreground, real gravitational-wave event, is shown as a golden star and is the coincidence between three detectors with a single detector trigger in each which falls within the allowed light travel time of the detector network. The dashed black box indicates a time slide being made where two detectors have their data shifted by an interval greater than the allowed light travel time window to produce events which can only be caused by non-astrophysical noise sources. An example of a background event is shown as the three black stars in the blue box on the right image. Taken from [? ].

the triggers from the second detector after shifting the time in the second detector by greater than the light travel time window. This ensures that any coincidences made cannot possibly have been due to a real signal because the coincidences will have occurred outside the light travel time window. An example of a background coincidence by these time slides for the three detector case can be seen in Figure ??.

The time sliding procedure is repeated many times to produce a large sample of false-alarm coincidences, which are used to compute the false-alarm rate as a function of the ranking statistic.

To measure the significance of each candidate event, we assign a p-value. In our pipeline, the p-value of a candidate event is the probability that there are one or more false-alarm events that have a ranking statistic value greater than or equal to the ranking statistic of the detector value,  $p_b = P(R_{FA} \geq R_{CE})$ . The p-values of candidate events are calculated under the null hypothesis that *all* triggers are seen due to noise. To confidently claim the candidate event as real, we must demonstrate that the null hypothesis given the candidate event ranking statistic value is highly improbable (the p-value is small).

We can do this by measuring the number of noise background events,  $n_b$ , that have a ranking statistic value higher than the candidate event. If we do this for all ranking statistic values we can build up a mapping of ranking statistic to false-alarm rate. The function  $n_b(R)$  gives the number of background events with a ranking



statistic value higher than  $R$ , the ranking statistic value. The probability that one or more background events are found above  $R$  given the observing time  $T$  and the background time  $T_b$  is [? ]

$$p(\geq 1 \text{ above } R | T, T_b)_0 = 1 - \exp \left[ \frac{-T(1 + n_b(R))}{T_b} \right]. \quad (2.38)$$

The background time will equal  $T_b = T^2/\delta$  where  $\delta$  is the time-slide interval. We can produce a very large amount of background data from a relatively small period of observing data, fifteen days of coincident data with a time-slide interval of 0.1 seconds allows us to measure false-alarm rates of 1 in 200,000 years.

Finally, we can express the mapping of false-alarm rate to ranking statistic as

$$\text{FAR}(R^*) = \int r_n(\vec{\kappa}) \Theta(R(\vec{\kappa}) - R^*) d^n \vec{\kappa}, \quad (2.39)$$

where  $\Theta$  is the Heaviside step function

$$\Theta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0, \end{cases} \quad (2.40)$$

and the false-alarm rate at the ranking statistic value  $R^*$  is being calculated by integrating over all possible background events and summing up the events that have a ranking statistic greater than or equal to  $R^*$ , the ranking statistic threshold. The optimal ranking statistic will maximise the expected number of coincident events due to signals recovered above  $R^*$ .

## 2.3 Gravitational-wave search pipelines

We can develop the idea of a gravitational-wave search pipeline by describing the structure of the PyCBC [? ] search pipeline and how gravitational-wave events are identified from the initial gravitational-wave data using all the techniques described previously in this chapter.

### 2.3.1 PyCBC

The flow chart describing the structure of the PyCBC pipeline has been taken from the PyCBC paper and can be seen in Figure ?? . We follow this flowchart in our description of the search pipeline.

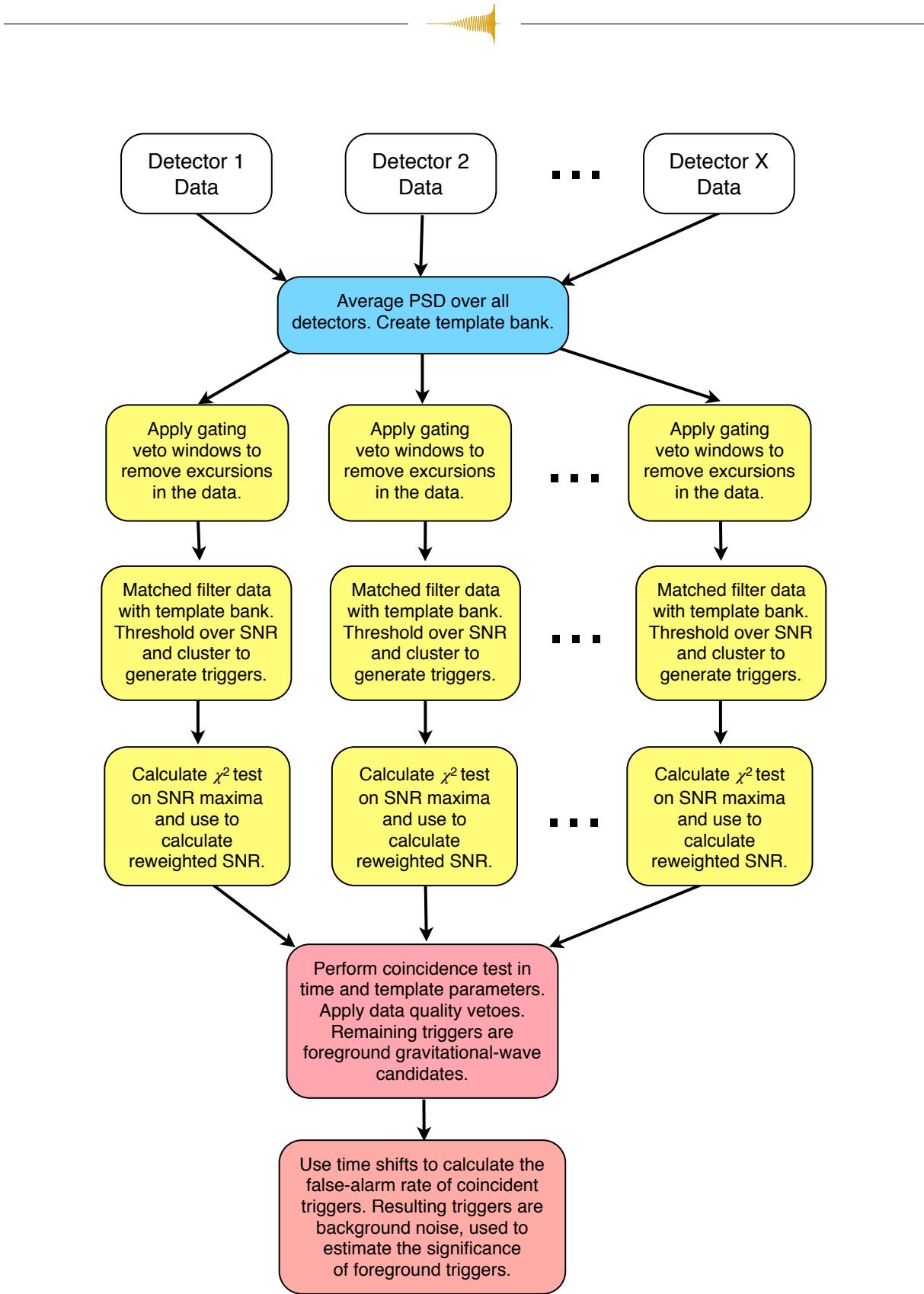


Figure 2.4: The structure of the PyCBC search pipeline for gravitational waves. Taken from [? ].



## Template bank generation

A key consideration for gravitational-wave search pipelines is computational cost. A greater computational cost requires more time to analyse gravitational-wave data, alongside greater monetary cost and carbon emission cost. The dominant cost for modelled searches is matched filtering. Therefore, the number of templates in the template bank is proportional to the computational cost of the search. The number of templates is tuned to balance a minimal loss in matched filter SNR with the computational cost.

The template bank in the PyCBC search is shared between all detectors to allow a coincidence requirement of triggers sharing the same template between detectors. The template bank is generated given a power spectral density (PSD) and therefore we must create a PSD that is averaged over time for all detectors. The PSD is measured every 2048 seconds independently in each detector using Welch's method and taking the median average, this gives  $N$  power spectra  $S_n$  for each detector. The harmonic mean PSD for a single detector across all analysis time is then found

$$S_n^{\text{harmonic}}(f_k) = N / \sum_{i=1}^N \frac{1}{S_n^i(f_k)}, \quad (2.41)$$

and using the same method we average across all detector harmonic mean PSDs to get an estimated PSD across all time and detectors to use for template bank generation. This PSD estimate only need to be re-generated when the detector's noise PSD has changed significantly, which will typically only happen when the detectors are physically changed or upgraded.

The PyCBC template bank is placed with a minimum match of 0.97 between any signal and the template bank. The loss in sensitivity due to this minimum match limit is equal to the minimum-match cubed and is  $\simeq 10\%$ . The template bank covers a four dimensional parameter space of two component masses and aligned spins. The PyCBC template bank was generated using a combined geometric-based aligned-spin algorithm [? ?] and a stochastic algorithm [? ?], described in [? ].

A third observing run template bank can be seen plotted in the  $m_1 - m_2$  parameter space in Figure ?? which contains 428,725 unique templates.

## Matched filtering and clustering

The matched filter between template bank and data is performed on each block of data to create a list of single detector triggers. Triggers are only stored if their matched filter SNR is greater than 5.5. Suppose we have a trigger with an SNR of 10.0, according

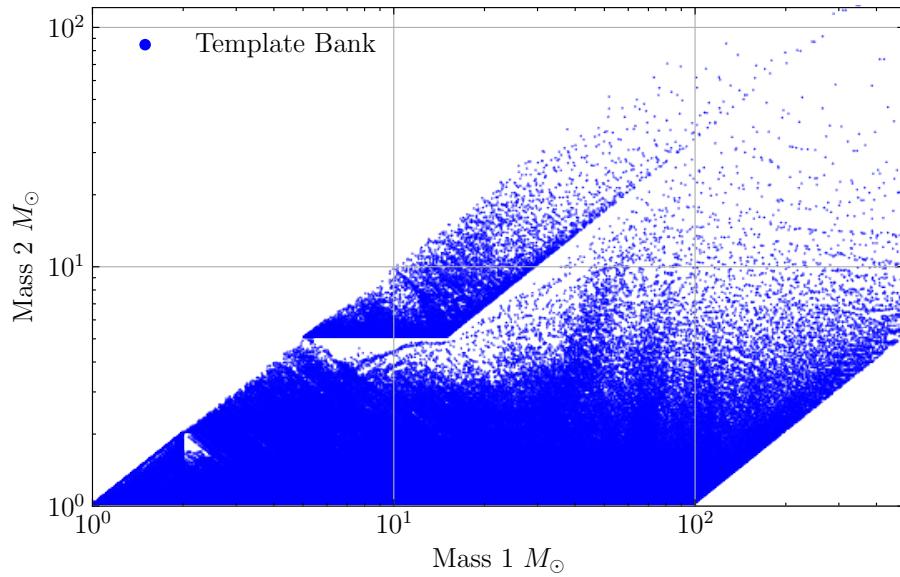


Figure 2.5: An example of a template bank used by the PyCBC search pipeline that has been generated from a single block of data and has been created in the four dimensional component mass and aligned spin parameter space. It consists of 428,725 templates.

to our template bank minimum match we should have at least one additional trigger (and indeed have many) above the SNR threshold for a nearby template. To prevent the recording of multiple triggers from different templates for the same candidate event, we use a *time clustering* algorithm which selects and keeps only the highest SNR trigger in a predefined time window. Another clustering algorithm, is employed to trigger across the template bank to prevent the triggering of a loud glitch in one region of the template bank from subduing a real signal trigger in a completely different region.

### Signal consistency test

The triggers then have the  $\chi^2$  test [? ] applied to ensure consistency in the signal power distribution. The  $\chi^2$  test values are calculated for each SNR trigger found by the template bank (Equation ??) and used to re-weight SNR values to obtain new SNR values for each trigger. An example of the SNR and  $\chi^2$  time series from a loud signal can be seen in Figure ??.

### Coincidence tests

Single detector triggers are combined and evaluated based on their trigger times and template parameters, triggers that have been found within the maximum light travel time between the two detectors (10 ms for LIGO-Hanford and LIGO-Livingston)

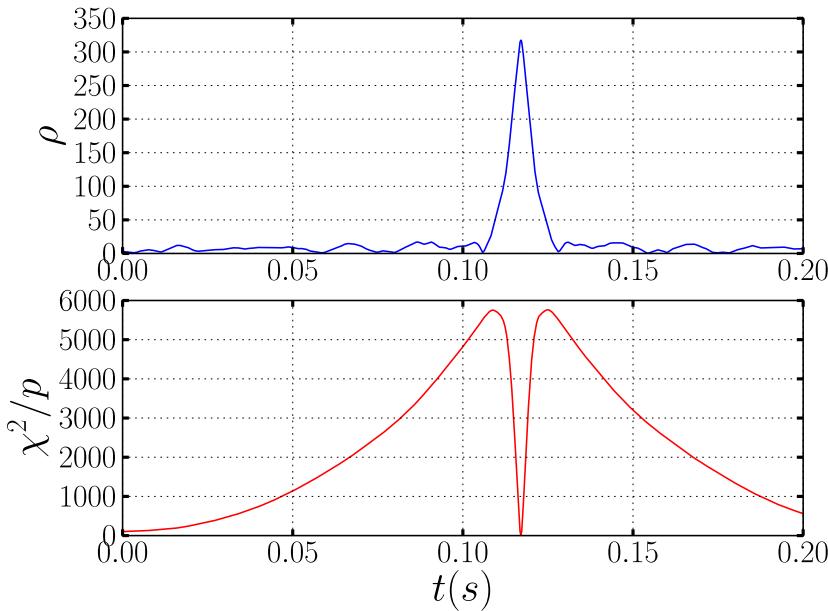


Figure 2.6: An SNR time series and  $\chi^2$  time series for a simulated compact binary coalescence injected into S5 [? ] data with an SNR of 300. Taken from [? ].

with an exactly matching template can be considered to originate from the same gravitational-wave signal [? ]. The result of this process is a list of coincident triggers found with SNR above the threshold in two or more detectors, with exact template parameters across detectors.

### Generate background, rank triggers, establish significance and estimate search sensitivity

The ranking statistic and false-alarm rates are calculated for each of these triggers as described in the previous section, using a quadrature sum of the reweighted SNR in each detector. The sensitivity of the search pipeline can be evaluated by injecting many simulated gravitational-wave signals into the data and recovering them with the search pipeline. The most realistic injections are hardware injections [? ? ] in which an actuator exerts a force on the end test mass to simulate the response of a real gravitational-wave signal. These injections are rarely performed because the data is now contaminated and cannot be used to search for real gravitational-wave signals. The injections used for large scale injection campaigns are software injections, in which the gravitational-wave signal is added to the data. These injections are added to add detectors with the correct time, phase, and amplitude differences and are placed so as to probe the full population of potential gravitational-wave events. The injection



campaigns are a very powerful tool that can reveal search pipeline sensitivity, and we have used them extensively throughout this thesis to test the sensitivity increase of changes made to search pipelines.



# 3

---

## Detector Characterisation

---

*“This is the decompress before the intermission”*

— Drake

In this thesis, we have made improvements to the search for gravitational waves by improving the quality of the detector data and understanding the noise background of the detectors.

This chapter is laid out as follows: we briefly repeat the state of the current gravitational-wave detector network in Section ??, in Section ?? we define the research area of detector characterisation, the systematic sources of noise faced in the detectors are described in Section ??, some common detector data noise transients in Section ?? and finally, the detector characterisation tools most commonly used and referred to throughout this thesis in Section ??.

### 3.1 Gravitational-wave detectors

Gravitational-wave astronomy in the present day is performed by ground-based detectors at different locations globally: the LIGO-Livingston and LIGO-Hanford detectors are located in the United States of America, the Virgo interferometer in Italy, and the KAGRA telescope in Japan. We will primarily reference the LIGO detectors in the United States of America when discussing the challenges faced in detector characterisation.



As described in Chapter ??, the LIGO detectors work on the principle of laser interferometry to detect gravitational-wave signals. The sensitivity [?] of gravitational-wave detectors is limited by fundamental sources of noise [?] such as seismic noise [?], thermal noise [?] and quantum noise [?]. Additionally, we have non-Gaussian noise [?], which manifests as transient noise bursts in the data, which can contribute to false positives in gravitational-wave searches and the obscuring of real gravitational-wave signals [??].

### 3.1.1 Detector analysis

Systematic sources of noise are the dominating limitation in the sensitivity of ground-based detectors. The sources of noise with the greatest contributions are

- Seismic noise [?]: the vibrations of the ground causes a coupled vibration motion in the mirrors of the interferometer. Higher frequency seismic noise (1 – 10Hz) is often caused by local ground movements typically anthropogenic in nature, or greater magnitude earthquakes [?]. Lower frequency seismic noise (< 1Hz) is caused by larger-scale geophysical processes such as ocean waves, tidal effects, or smaller local earthquakes [?]. Seismic noise can be mitigated with advanced suspension systems to isolate mirrors and end-station components from the ground to prevent coupling to seismic motion [?].
- Thermal noise [?]: the thermal energy of interferometer components will cause them to vibrate. This noise can be reduced with new mirror materials with resonant frequencies outside the sensitive regions or with cryogenically cooled mirrors such as those used by KAGRA [?].
- Shot noise [?]: the distribution in the number of photons observed by the interferometer photodetector in any time interval is Poisson distributed, and the error in the photon count places limits on the sensitivity. The error is proportional to the square-root of the count ( $\sigma \propto \sqrt{\lambda}$ ) therefore, the simplest way to reduce this error is to increase the laser power and increase the photon count in the time interval.
- Radiation pressure [?]: the pressure exerted by the photons hitting the end mirrors will decrease search sensitivity. To reduce this noise, the laser power can be reduced; however, this will increase the shot noise.

It can be seen that the laser power must be calibrated to balance the impact of increasing the photon count, which will decrease shot noise but simultaneously in-

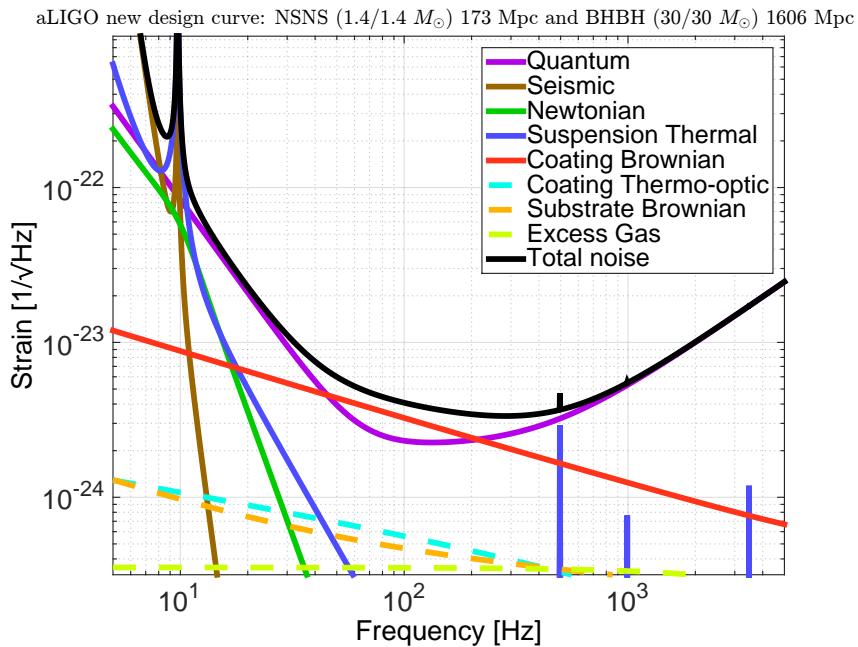


Figure 3.1: The Advanced LIGO [? ] strain sensitivity as a function of frequency (black solid line), accompanied by the systematic noise sources which limit the sensitivity of the detector. Taken from [? ].

crease radiation pressure noise. A figure displaying these systematic noise sources and their limitations in the LIGO detector case can be seen in Figure ??.

To monitor detector sensitivity, the binary neutron star inspiral range is calculated. This is the range at which a binary neutron star signal, with both components having a mass of  $1.4 M_{\odot}$ , will be detected given the characteristic noise (PSD) of the data. The PSD is calculated by taking an average of the detector noise in the previous minute [? ]. Figure ?? shows an example of the binary neutron star range for a day of LIGO-Hanford during the fourth observing run.

### 3.1.2 Noise transients

Noise transients, commonly referred to as glitches, are short duration bursts of noise found in gravitational-wave data. The systematic sources of noise described in Section ?? limit the sensitivity of the detector to a certain frequency range, glitches appear within this sensitivity frequency range. The specific noise transients investigated by detector characterisation are non-Gaussian noise artefacts that have the ability to obscure [? ] or mimic gravitational-wave signals [? ], producing false-alarms in our gravitational-wave search pipelines. Understanding these glitches is crucial for gravitational-wave detection; they have the potential to reduce the sensitivity and

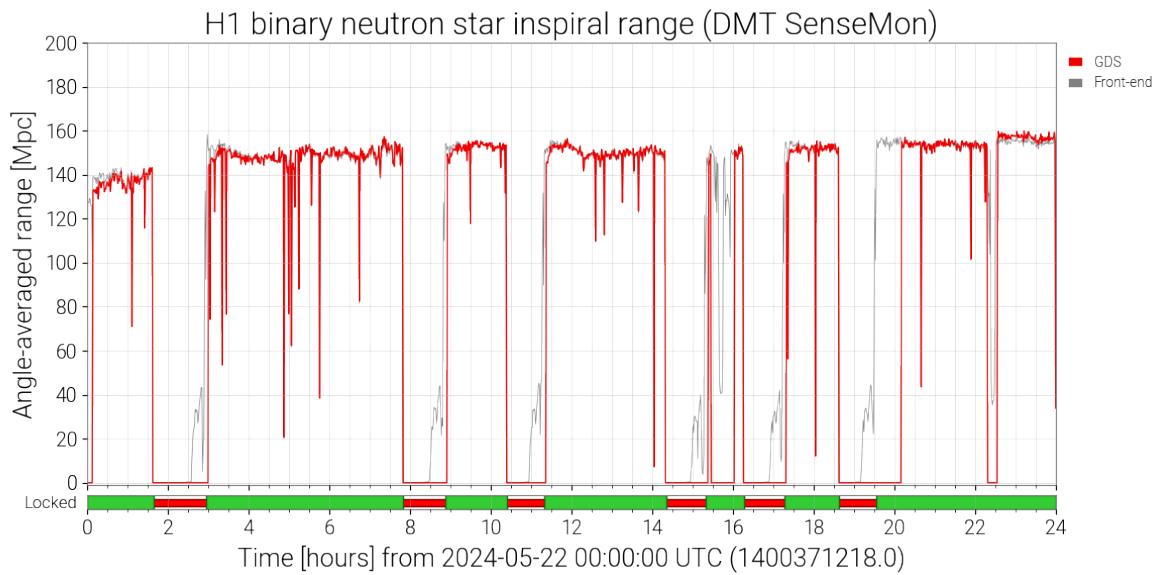


Figure 3.2: The LIGO-Hanford binary neutron star inspiral range for the 22nd May 2024, created using GWSumm [? ] and taken from the LIGO Summary Pages-<https://summary.ligo.org/>, please see [https://gwosc.org/detector\\_status/](https://gwosc.org/detector_status/) for the public summary pages.

hinder the reliability of the detectors.

There are at least 27 [? ? ] different classes of glitch which all manifest with different durations (typically in the millisecond to second range), amplitudes, and glitch morphology. Glitches populate the whole sensitive frequency range of the detectors, with some glitches being broadband, affecting up to the whole frequency bandwidth, to others being narrowband and affecting only specific frequency ranges. Glitches are commonly characterised and studied in 2-dimensional time-frequency representations of the one-dimensional strain time series that the detector outputs. The typical time-frequency representation used is the OmegaScan [? ] (referred to as ‘Omega scan’ or ‘Q-scan’) which is described later in this chapter.

Glitches originate from either environmental sources, instrumental sources, or a coupling of the two. As mentioned previously, seismic noise limits the sensitivity of the detectors below  $\sim 10$  Hz, but the coupling of seismic motion into interferometer components can cause one of the more common glitches—scattered light. Other environmental noise sources are heavy winds, lightning, and human activity (traffic, construction, trains). A historical glitch identified at LIGO-Livingston was caused by an air conditioning compressor cycling, these glitches were seen by both a magnetometer and the gravitational-wave strain channel. Other instrumental glitches might be caused by mirror suspensions, electronics, control systems or laser fluctuations.

The most common glitches are: blips [? ], scattered light [? ] and whistles [? ],

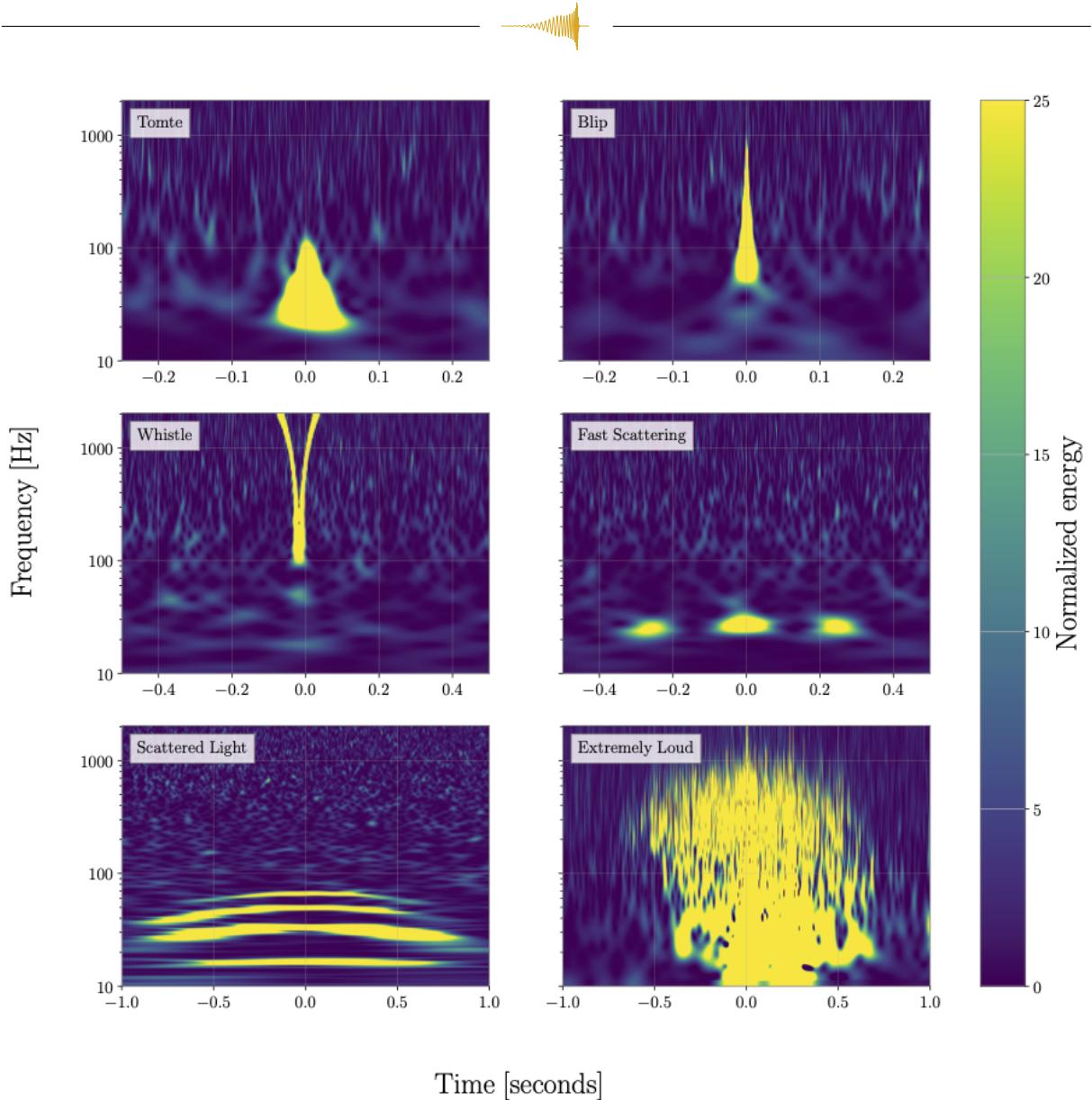


Figure 3.3: Six classes of glitch commonly found in LIGO detector data. Taken from [? ? ].

Omega scans of these glitches can be seen in Figure ???. The sources of these glitches have been studied. For example, whistles are caused by the beating of radio frequencies in the detector; however, some glitches come from unknown sources, such as blips. Scattered light has been found from many sources in the detector and great effort has been made to reduce the presence of scattered light in the data [? ] but as of the fourth observing run some scattered light still remains.

A number of algorithms and tools have been developed to detect and characterise noise transients in gravitational-wave data [? ? ? ? ? ? ? ? ? ? ? ? ]. The primary aim of these tools is the identification of glitches and analysis of properties of the glitches (amplitude, frequency, duration, waveform shape) to determine the source of



the noise. Once the noise source has been identified, improvements to the detector can be made to eliminate it. Other tools focus on modelling noise transients for the purpose of subtracting them from affected gravitational-wave data [? ? ? ?].

Another tool used in the identification and classification of glitches is the thousands of auxiliary channels alongside the main strain channel [? ], which monitor the many subsystems of the detector and can be used to find noise correlations between the different parts of the instrument [? ]. The auxiliary channels which are not sensitive to gravitational waves are very useful in identifying any correlations between noise in the gravitational strain channel and noise also seen in another channel which cannot observe gravitational waves, an example of this is the magnetometer recording the same glitch as seen in the gravitational-wave strain channel when the air conditioning compressor was cycling at LIGO-Livingston [? ].

## 3.2 Detector characterisation and calibration

Detector characterisation is a field of study which assess and improves the performance of gravitational-wave detectors to ensure accurate and reliable detection of gravitational waves. Gravitational-wave detector calibration determines how the response of the detector translates to astrophysical signals, ensuring accuracy in extracted astrophysical information. Detector characterisation is responsible for identifying and mitigating the many sources of noise that can obscure or mimic real signals. The optimisation of detector sensitivity is also studied, whereby different designs and components can be fine-tuned to increase sensitivity. Another responsibility of detector characterisation is the monitoring of detector performance in real-time to assess sensitivity changes.

Overall, effective detector characterisation ensures that gravitational-wave observatories can operate at maximum potential and provide accurate and reliable observations of gravitational waves. The gravitational-wave detector organisations have dedicated working groups for detector characterisation and calibration [? ? ]. In this thesis, chapter ?? is a direct contribution to the field of detector characterisation with a contribution to the LIGO detector characterisation working group.

The detector characterisation group has the important task of verifying data quality when a gravitational-wave signal is detected by live detection pipelines, giving the confirmation that the gravitational-wave signal isn't a false alarm potentially caused by noise transients or poor data quality immediately surrounding the gravitational-wave signal.



### 3.2.1 Detector characterisation tools

**Omega scans** are two-dimensional time-frequency representations of gravitational-wave data used very commonly to display gravitational-wave data in a human interpretable fashion, which contains more information than a simple spectrogram and can reveal time-frequency relationships that are virtually impossible for humans to see in the time series. Example Omega scans can be seen in Figure ?? for a number of glitches.

Detector characterisation uses Omega scans to highlight glitches which can have sharp and localised features in the time-frequency plane. Omega scans calculate the amplitude of strain power at each pixel using the *Q-transform* [?], where for each point in time and frequency a number of wavelets are created with a varying *quality factor* ( $Q$ ).  $Q$  is calculated using

$$Q = \frac{f_c}{\sigma_f}, \quad (3.1)$$

where  $f_c$  is the central frequency and  $\sigma_f$  is the bandwidth of a wavelet. The bandwidth has a uncertainty relation with the duration,  $\sigma_t$  of the wavelet

$$\sigma_t \sigma_f \geq \frac{1}{4\pi}, \quad (3.2)$$

and so a wider frequency bandwidth will lead to a shorter duration for the wavelet, balanced by the Q-factor. The Q-transform of all these wavelets with the gravitational-wave strain is taken

$$H(\tau, f, Q) = \int_{-\infty}^{\infty} h(t) w(t - \tau, f, Q) dt, \quad (3.3)$$

and for each pixel the Omega scan algorithm will calculate the power of every wavelet,  $|H(\tau, f, Q)|^2$ , and select the one with the greatest amount of power. As an example, a short-duration glitch (like a blip) would be better tiled with multiple low-frequency bandwidth, high-Q tiles which will resolve the glitch in greater definition than a low-Q tile which captures the whole glitch in one tile.

**GravitySpy** is a citizen science machine learning tool for classifying glitches found in gravitational-wave data [?]. GravitySpy has been trained by volunteers at the project website: <https://www.zooniverse.org/projects/zooniverse/gravity-spy>, hosted on the Zooniverse [?].

GravitySpy has uploaded more than 1.4 million of Omega scans of gravitational-wave data which have been found to contain bursts of power which could be caused



by glitches. The Omega scans are published on the GravitySpy project and volunteers are given these images and the option for which glitch they think is contained within the image. Once these images have been classified, the machine learning algorithm is trained, and it can go further to classify the images without the need of volunteers [? ].

GravitySpy has been run on data from the second and third observing runs, where it was able to classify 24 different glitch categories. The downsides of GravitySpy are the constantly changing noise background of the gravitational-wave detectors and the new glitch types emerging which need to be trained on with volunteering again [? ].

**Data Quality Veto**s are a flag applied to periods of data which contain potential data quality issues [? ]. These periods of data are identified using the  $\sim$ 200,000 auxiliary channels [? ], an example is a significantly elevated transient noise rate in the strain channel five days prior to GW150914 [? ] which was traced back to the 45 MHz electro-optic modulator driver system used to generate optical cavity control feedback signals [? ]. The noise caused by this channel was given a category 1 veto data quality flag and removed 2.62% of the total coincident time from the analysis period. While there is a significant reduction in the number of high SNR triggers that would've been found during these times, these flags fallen out of favour in the fourth observing run due to removing large amounts of data.

# 4

---

## ArchEnemy

---

*“One less enemy within.”*

— Swain, League of Legends

This chapter reproduces the published work carrying the name “*ArchEnemy: Removing scattered-light glitches from gravitational wave data*”, in CQG [? ] with changes for formatting consistency and no changes to content. I am the first author of this publication and was responsible for the development of the scattered-light glitch model,  $\chi^2$  test and glitch search pipeline and I then tested the changes using a PyCBC Offline search injection study to evaluate the glitch search and subtraction tool ArchEnemy.

### 4.1 Introduction

The Laser Interferometer Gravitational-Wave Observatory (LIGO) [? ] and Virgo [? ] collaborations made the first observation of gravitational waves in September 2015 [? ]. The detection established the field of gravitational-wave astronomy and a global network of gravitational-wave detectors, now joined by KAGRA [? ], has allowed for the detection of approximately 100 gravitational-wave events [? ? ? ? ? ? ? ].

The detection of gravitational waves is made possible by both the sensitivity of the detectors and the search pipelines [? ? ? ? ? ] which analyse raw strain data from the output of the detectors and identify observed gravitational-wave signals.

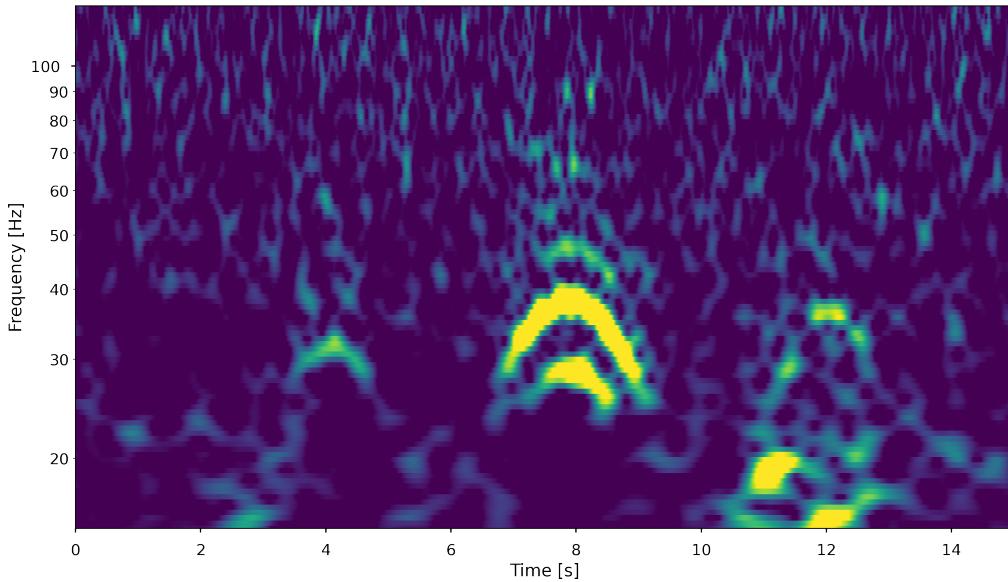


Figure 4.1: An Omega scan [? ] of gravitational-wave data containing an example of a scattered-light glitch. Scattered-light glitches are characterised by a symmetric arch-like pattern. Multiple scattered-light glitches can be seen at the 4, 8 & 12 second marks as well as multiple harmonic glitches at 8 seconds.

One of the problems that these search pipelines must deal with is the fact the data contains both non-stationary noise and short duration ‘glitches’ [? ? ? ] where noise power increases rapidly. Glitches are caused by instrument behaviour or interactions between the instrument and the environment [? ? ] and glitches reduce the sensitivity of the detectors [? ], can potentially obscure candidate gravitational-wave events [? ] and can even mimic gravitational-wave events [? ? ].

Different classes of glitches have been characterised using tools such as Gravity Spy [? ? ]. Of the 325,101 glitches classified by Gravity Spy in the third observing run of Advanced LIGO [? ] with a confidence of 90% or higher, 120,733 (32.1%) were classified as “Scattered Light”. Scattered-light glitches occur in the 10-120Hz frequency band [? ] which coincides with the frequency band where we observe the inspiral and merger signatures of compact binary coalescences (CBCs). Scattered-light glitches are characterised by an arch-like pattern in a time-frequency spectrogram of the detector output, as seen in Figure ???. Scattered-light glitches occur when laser light in the interferometer is scattered from the main optical path by components within the detector. The motion of these components is coupled to seismic motion, inducing a phase shift on the light being scattered as the surface moves back and forth. This scattered light then recombines with the main laser, producing scattered-

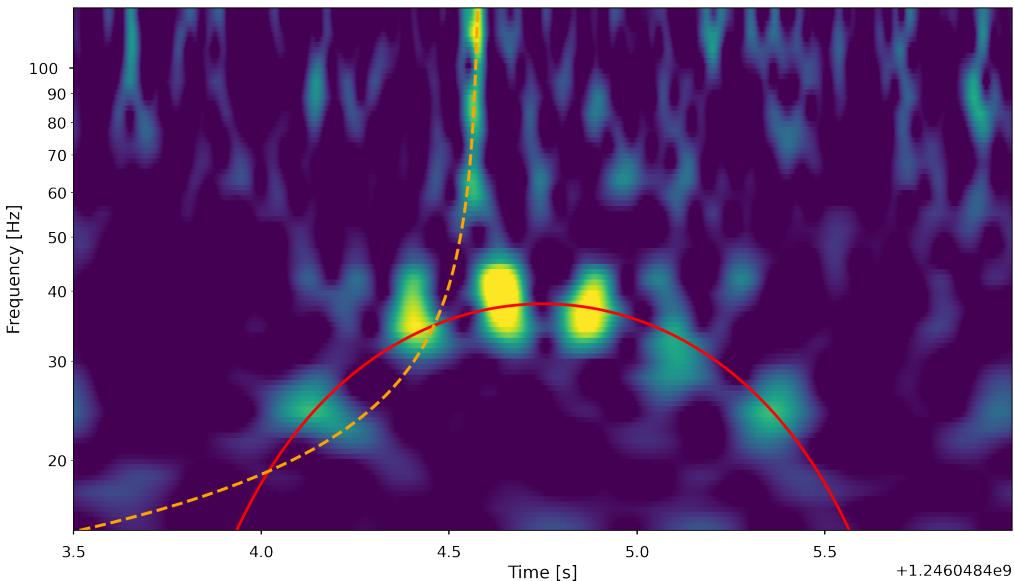


Figure 4.2: GW190701\_203306, a gravitational-wave event coincident with a scattered-light glitch in the data from the LIGO Livingston observatory. The orange dashed track shows the inferred time-frequency evolution of a gravitational-wave event produced by a compact binary coalescence, the red solid line is an overlaid track of the approximate location of the coincident fast scattering glitches.

light glitches in the data. The surfaces from which scattered-light glitches originate have been objects on optical benches such as lenses, mirrors and photo-detectors [? ].

Scattered-light glitches have been a significant problem when observing CBCs. As an example, GW190701\_203306 was coincident with a scattered-light glitch, as shown in figure ?? [? ], requiring subtraction from the data before the event could be properly characterised [? ]. A further 7 candidate events were found to be in coincidence with scattered-light glitches in the third observing run [? ]. For this reason, it is important to reduce the effect of scattered-light glitches in the detectors and gravitational-wave search pipelines. Scattered-light glitches occur as single or multiple glitches and can appear rapidly in time and simultaneously in frequency (see figure ??), which we refer to as harmonic glitches.

The most obvious way to remove the impact of glitches is removing the mechanisms which produce the glitches in the observatories. This has been investigated in previous works [? ? ? ? ? ? ? ? ], which focus on identifying the surfaces in which light is being scattered from and then mitigating the scattering by reducing the reflectivity of the surface, seismically isolating it or relocating it.

An alternative method for reducing scattered-light glitches, known as “RC track-

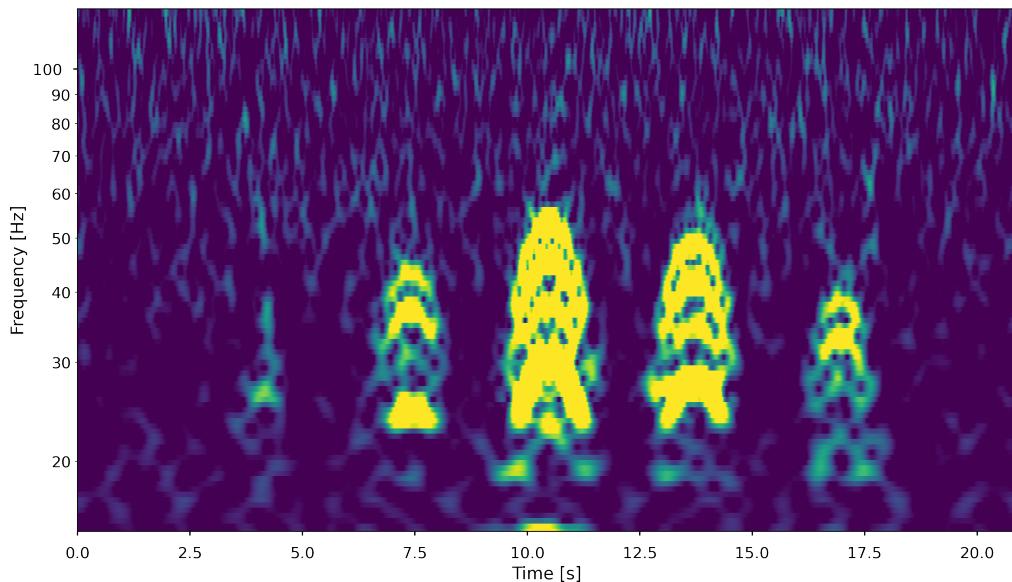


Figure 4.3: An Omega scan [? ] of gravitational-wave data containing multiple examples of scattered-light glitches. Here we see multiple scattered-light glitches repeating periodically in a 20 second period of time. Harmonic scattered-light glitches are also seen in multiple stacks, the harmonic glitches share *time period* values and the *glitch frequency* values are  $n$ -multiples of the lowest frequency harmonic within the stack.



ing”, was implemented in the Advanced LIGO observatories in January 2020 [? ]. The Advanced LIGO detectors employ a quadruple pendulum suspension for the test masses, where two chains suspend four masses in this suspension system, one for the test mass optic and the other for the reaction mass. The reaction mass is used to impose a force upon the test mass, and a significant source of scattered-light glitches was the large relative motion between the test mass chain and the reaction chain. To mitigate this effect, the relative motion between the end test mass and the reaction mass needed to be reduced. This was achieved by ensuring the two chains are moving together by applying force to the top stage of the quadruple suspension system in Advanced LIGO. The implementation of RC tracking represented a decrease from  $0.01\text{s}^{-1}$  to  $0.0001\text{s}^{-1}$  and  $0.0072\text{s}^{-1}$  to  $0.0012\text{s}^{-1}$  in the number of scattered-light glitches detected by Gravity Spy for LIGO-Hanford and LIGO-Livingston, respectively.

While methods for preventing scattered-light glitches have been developed and have shown success, they have not been able to remove the problem of scattered-light glitches from the data. Additionally, as the detectors continue to be upgraded and increase in sensitivity, new sources of scattered-light glitches will continue to appear. Identifying these new sources and mitigating their effects can take many months, during which time the detectors are taking in data which might be affected by the presence of scattered-light glitches. Therefore, it is not realistic to believe that analyses will be able to regularly run on data that does not contain scattered-light glitches and this motivates us to develop a technique for mitigating the impact of these glitches when trying to identify CBCs in gravitational-wave data.

In this work, we present a new method for identifying and removing scattered-light glitches from gravitational-wave data in advance of running searches to identify CBCs. We first introduce a method for identifying when scattered-light glitches are present in detector data, through the creation of a new modelled search for scattered-light glitches, similar to how we search for gravitational waves using matched filtering. We can model scattered-light glitches, generate a suitable set of glitch waveforms and perform a matched filter search on detector data. We then subtract identified glitches from the data to increase detector sensitivity. The detector data isn’t Gaussian and stationary, so the matched filter does have the potential to identify non-scattered-light glitches, and potentially even gravitational-wave signals, as scattered-light glitches. To prevent this we also demonstrate a new scattered light  $\chi^2$  test, which can distinguish between scattered-light glitches, and other glitches—and gravitational-wave signals—in the data.

We begin by reviewing previous research and describing the formulation of the waveform model used for characterizing scattered-light glitches in section ?? . In sec-



tion ?? we introduce the various techniques used in the search to identify scattered-light glitches in gravitational-wave data and the results of the scattered-light glitch search. In section ?? we describe the results of a “glitch-subtracted” gravitational-wave search and any increases in sensitivity. We conclude in section ?? and discuss the implementation of this method in future observing runs.

## 4.2 Scattered light

To identify scattered-light glitches in gravitational-wave data requires an accurate model of scattered-light glitches. This section details the derivation of the model we will use for generating our scattered-light glitch filter waveforms, along with its parameterisation.

### 4.2.1 Modelling scattered-light glitches — a review

Our model for scattered-light glitches draws heavily from [?], and we briefly review the main details of the model presented there. In [?], the authors construct a model to accurately predict the increase in noise due to scattered light during periods of increased micro-seismic activity. The model in [?] is constructed from parameters related to physically measurable properties of the detector such as the mirror transmission factor,  $T$ , the finesse of the Fabry–Pérot cavity,  $F$ , or the wavelength of the light,  $\lambda$ .

They define the amplitude of the additional beam produced by light scattering off of a surface as

$$A_{sc} = A_0 T \sqrt{\frac{2F}{\pi}} \sqrt{f_{sc}} e^{i\phi_{sc}}, \quad (4.1)$$

where  $A_0$  is the amplitude of the light resonating in a Fabry–Pérot cavity,  $f_{sc}$  is the fraction of the optical power scattered back into the main beam and  $\phi_{sc}$  is the phase angle modulated by the displacement of the scattering optics, defined as

$$\phi_{sc}(t) = \frac{4\pi}{\lambda} (x_0 + \delta x_{opt}(t)), \quad (4.2)$$

where  $\delta x_{opt}$  is the displacement of the scattering surface and  $x_0$  is the static optical path.

The total amplitude of the beam inside the arm is given by  $A_{tot} = A_0 + A_{sc}$ , with a phase angle equal to the phase noise introduced by the scattered light  $\delta\Phi = \frac{A_{sc}}{A_0} \cdot \sin \phi_{sc}$ . The noise introduced by the scattered light,  $h_{sc}$ , can be approximated through the relationship  $\sin(\phi + \delta\phi) \approx \sin \phi + \cos \phi \cdot \delta\phi$  for small  $\delta\phi$ , and can be



expressed as

$$h_{sc}(t) = G \cdot \sin\left(\frac{4\pi}{\lambda}(x_0 + \delta x_{sc}(t))\right), \quad (4.3)$$

where  $G$  is the *coupling factor*, defined as  $K \cdot \sqrt{f_{sc}}$  where

$$K = \frac{\lambda}{4\pi} \frac{T}{\sqrt{2F\pi}}. \quad (4.4)$$

The displacement of the scatterer when presenting with oscillatory motion is then given as

$$\delta x_{sc}(t) \simeq A_m \sin(2\pi f_m t), \quad (4.5)$$

where  $f_m$  is the frequency-modulated signal with modulation index  $m = A_m \frac{4\pi}{\lambda}$  and where  $A_m$  is the amplitude of the  $n$ th harmonic. Finally, Equation ?? can be simplified when considering only small bench motion, according to

$$h_{sc}(t) = G \cdot \cos \phi_0 \cdot \frac{4\pi}{\lambda} \cdot \delta x_{sc}(t) + h_{\text{offset}}. \quad (4.6)$$

## 4.2.2 Model

The model introduced in [? ] for the gravitational-wave strain noise introduced by scattered light uses a lot of knowledge about the detector state. The model used in this work will be more phenomenological in the parameterisation, allowing us to rely only on the characteristics of the glitches in the strain data and not knowledge of the detector configuration, especially in cases where this detector information might not be known. Each scattered-light glitch, as viewed in a spectrogram (see figure ??), has two easily identifiable features: the maximum frequency reached, *glitch frequency* ( $f_{gl}$ ); and the period of time the glitch exists in detector data, *time period* ( $T$ ). In addition, we can fully describe an artifact by defining an *amplitude* ( $A$ ), *phase* ( $\psi$ ) and *centre time* of the glitch ( $t_0$ ).

To formulate a model of scattered-light glitches in terms of these parameters, we simplify equation ??, treating the strain noise caused by scattered light as the sinusoidal function

$$h_{sc}(t) = A \sin\left(\frac{4\pi}{\lambda}x(t) + \psi\right), \quad (4.7)$$

where  $A$  is some constant,  $x(t)$  is the displacement of the scattering surface and  $\psi$  is an offset.

Scattered-light glitches are caused by the physical increase in the distance travelled by the light as a consequence of being reflected off of a surface. We assume a simple



harmonic oscillator for the surface displacement with frequency  $f_{rep}$

$$x(t) = B \sin(2\pi f_{rep} t), \quad (4.8)$$

which is directly related to the *time period*,  $T$ ,

$$f_{rep} = \frac{1}{2T}. \quad (4.9)$$

The *time period* of a scattered-light glitch only corresponds to half of a sinusoidal wave hence the multiplier of 2 on the denominator of equation ???. We calculate the instantaneous frequency of the strain noise [? ]

$$f(t) = \left| \frac{1}{2\pi} \frac{d}{dt} \left( \frac{4\pi}{\lambda} x(t) + \psi \right) \right|, \quad (4.10)$$

which we can use to predict the frequency of each scattered light glitch. We define the maximum frequency of each glitch (at the peak of the arch) as the glitch frequency,  $f_{gl}$ , which can be found by taking the maximum of Equation ??

$$f_{gl} = \max [f(t)], \quad (4.11)$$

giving an equation for the glitch frequency of a scattered light glitch

$$f_{gl} = B \frac{4\pi}{\lambda} f_{rep}. \quad (4.12)$$

Rearranging Equation ?? yields the expression  $\frac{4\pi}{\lambda} = \frac{f_{gl}}{f_{rep} B}$  and when combined with Equations ?? and ?? gives the final form of the strain noise induced by scattered light

$$h_{sc}(t) = A \sin \left( \frac{f_{gl}}{f_{rep}} \sin(2\pi f_{rep} t) + \psi \right), \quad (4.13)$$

where  $A$  and  $\psi$  are amplitude and phase parameters to be maximised over.

In [? ] they provide another term to describe scattered-light glitches which uses the instantaneous frequency as a function of time, allowing the identification of the correct amplitude at all frequencies. This term is due to radiation pressure coupling and is thought to be dominant at low frequencies. The relationship between our amplitude and the new amplitude depends on the power in the arm cavities and the signal recycling mirror reflectivity, which we do not consider in our model and so disregard this extra term.



### 4.2.3 Harmonics

As described in [? ], harmonic glitches appear at the same time with different *glitch frequencies*. A harmonic glitch is a glitch with a *glitch frequency* that is a positive integer multiple of the glitch frequency of the glitch in the stack with the lowest glitch frequency value. This lowest frequency glitch has the potential to appear below 15Hz and will be masked by other sources of noise and therefore cannot be seen. An example of harmonics can be seen in figure ??.

## 4.3 Identifying scattered-light glitches in gravitational-wave strain data

Equipped with our model for scattered-light glitches from the previous section, we now discuss how we identify and parameterize scattered-light glitches in a stretch of gravitational-wave data before we apply this to searches for CBCs in the next section.

### 4.3.1 Matched filtering

Given our model of scattered-light glitches, we can consider a realisation with specific values of the parameters discussed above. To identify glitches with these values of the parameterisation we use matched filtering, described in Section ???. Scattered-light glitches will take a range of values of the parameters describing them and we must be able to identify glitches anywhere in the parameter space. Following [?] we can analytically maximize over the *amplitude*, *phase* and the *centre time* glitch parameters in equation ?? exactly as described in Section ??.

### 4.3.2 Template bank

Our scattered-light glitch model is parameterised by 5 variables. As shown above, by maximizing over *phase*, *amplitude* and *time*, we can analytically maximize the SNR over 3 of these variables. However, the remaining two describe the intrinsic evolution of the glitch, and we must explicitly search over these parameters. To do this we create and use a “template bank” of scattered-light glitch waveforms, created such that it would be able to identify glitches with any value of our 2 remaining parameters, *glitch frequency* and *time period*.

To generate this template bank, a stochastic template placement algorithm [?] is used. This algorithm randomly generates a new template, places the template in



the existing template bank and calculates the “distance” (how similar two templates appear) between the new template and existing templates. If the new template is too close to an existing template it is discarded, otherwise, it is accepted into the bank. The density of the bank is dependent on the allowed distance between two templates. The dominant cost of the search for scattered-light glitches is matched filtering and is approximately linear in the number of templates, a larger template bank will find all the glitches with more accurate parameter values, but the computational cost of the search will be increased. The template bank generation alone does not constitute a significant computational cost in the search.

The distance function we have chosen to evaluate templates by is the match between two glitch templates. To calculate the match, we first normalize both templates such that the matched filter between a template and itself would be equal to 1. We can then compute the inner product between the two normalised templates to find their match

$$M = \max_{t,\psi}(\hat{a}|\hat{b}). \quad (4.14)$$

The value for the match,  $M$ , is bounded between 0 and 1, where a value of 0 indicates orthogonal waveforms and a value of 1 indicates identical normalised waveforms. The maximum match allowed between any two templates in our bank is 0.97, which implies that a fully converged stochastic bank would have at least one waveform in it with  $M \geq 0.97$  for any point in the space of parameters.

For our scattered-light glitch search, we generated a template bank with *time periods*  $\in 1.8\text{ s}-5.5\text{ s}$  and *glitch frequencies*  $\in 20\text{ Hz}-80\text{ Hz}$ . We chose to use the Advanced LIGO zero-detuned high-power sensitivity curve [? ]<sup>1</sup>. This allowed us to generate a template bank that contains 117,481 templates. We visualize the distribution of the templates as a function of *time period* and *glitch frequency* in Figure ??, observing a greater density of templates at higher frequencies and longer durations, indicating that templates in this region look less similar and therefore, more templates are needed to fully describe the parameter space.

---

<sup>1</sup>The zero-detuned high-power sensitivity curve is a broader noise curve than the O3 Advanced LIGO data that we identify scattered-light glitches in. However, this broader noise curve will result in *more* template waveforms than we need, and will therefore overcover the parameter space, rather than potentially miss scattered-light glitches.

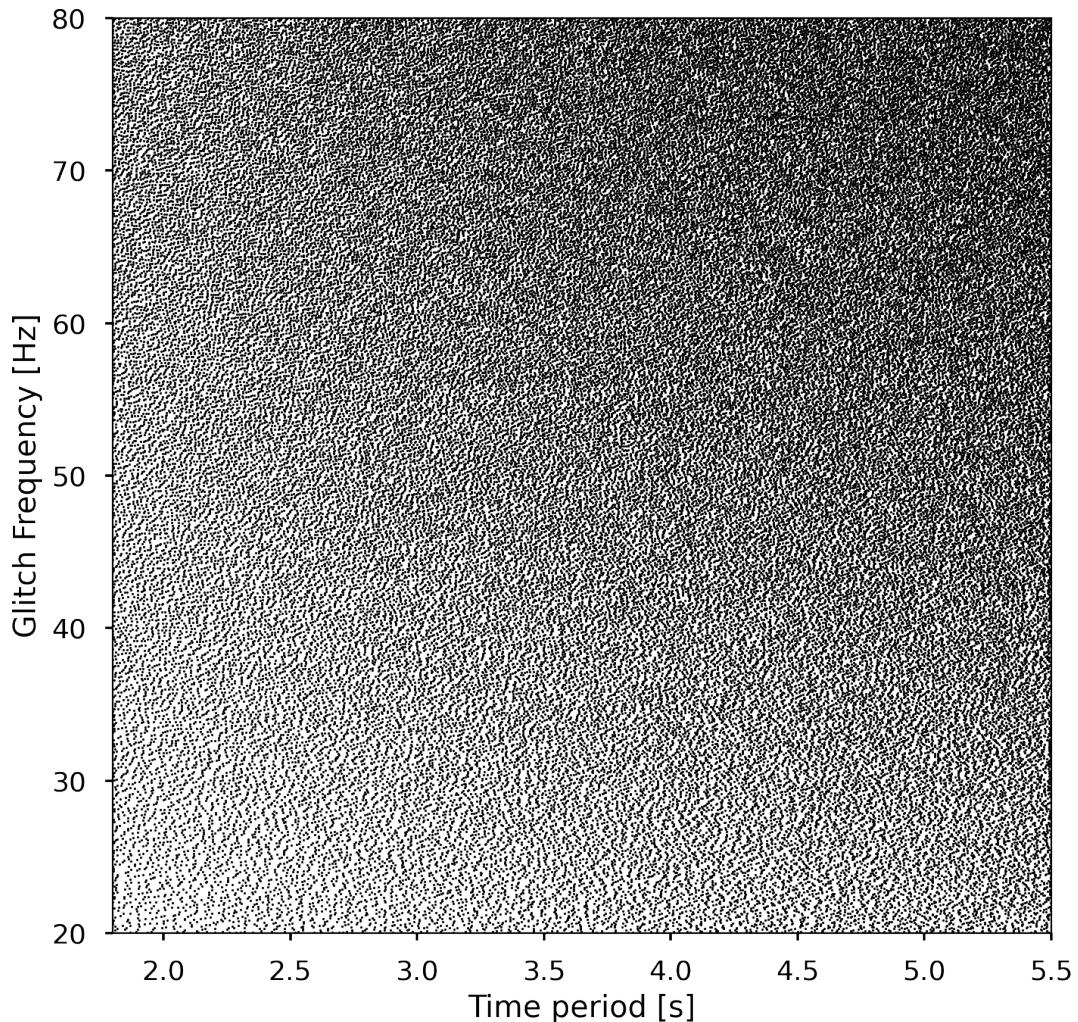


Figure 4.4: The template bank of scattered-light glitch templates (black points) used in the search for scattered-light glitches. The *glitch frequency* parameter values range from 20 Hz–80 Hz and the *time period* parameter values range from 1.8 s–5.5 s. This bank was created with a maximum match allowed between two templates of 0.97 and contains 117,481 templates.



### 4.3.3 Identifying potential scattered-light glitches in the data

We test our method by searching through gravitational-wave data from 2019-11-18 16:13:05–2019-11-25 22:11:09 for the LIGO-Hanford and LIGO-Livingston detectors. This time corresponds to the 25th period of data used in the LVK analysis of O3 data for CBCs [?] and is prior to the implementation of RC tracking [?] in these detectors. We only analyse data that is flagged as “suitable for analysis” on the Gravitational Wave Open Science Center [?]<sup>2</sup>. This corresponds to 5.96 days of analysable data for LIGO-Hanford and 5.93 days for LIGO-Livingston.

Equipped with our template bank, we now identify potential scattered-light glitches in the data. We matched filter all the data against all the templates, producing a SNR time series for every template in the bank. These SNR time series will contain peaks which, when above a certain limit, indicate a potential scattered-light glitch at a particular time. We retain any maxima within the time series that have a SNR greater than 8. However, as we do this independently for every template, we will identify multiple peaks for any given glitch, and we will also find peaks that correspond to other types of glitch, or even gravitational-wave signals. We discuss how we reduce this to a list of identified scattered-light glitches in the following subsections.

### 4.3.4 Scattered-light signal consistency test

To prevent the search for scattered-light glitches from misclassifying other classes of glitches, or gravitational-wave signals, we employ a  $\chi^2$  consistency test. The  $\chi^2$  discriminator introduced in [?] divides gravitational-wave templates into a number of independent frequency bins. These bins are constructed so as to contain an equal amount of the total SNR of the original matched filter between template and data. The  $\chi^2$  value is obtained by calculating the SNR of each bin, subtracting this from the expected SNR value for each bin and squaring the output. These values are summed for all bins and this value forms the  $\chi^2$  statistic,

$$\chi_r^2 = \frac{\chi^2}{\text{DOF}} = \frac{n}{2n-2} \sum_{i=1}^n \left( \frac{\rho}{\sqrt{n}} - \rho_{bin,i} \right)^2. \quad (4.15)$$

Here  $n$  is the number of bins,  $\rho$  is the SNR of the original matched filter between template and data, and  $\rho_{bin}$  is the value of the SNR found when matched filtering

<sup>2</sup>We note that in O3 *only* data suitable for analysis is released, so we simply analyse all the publicly available data.

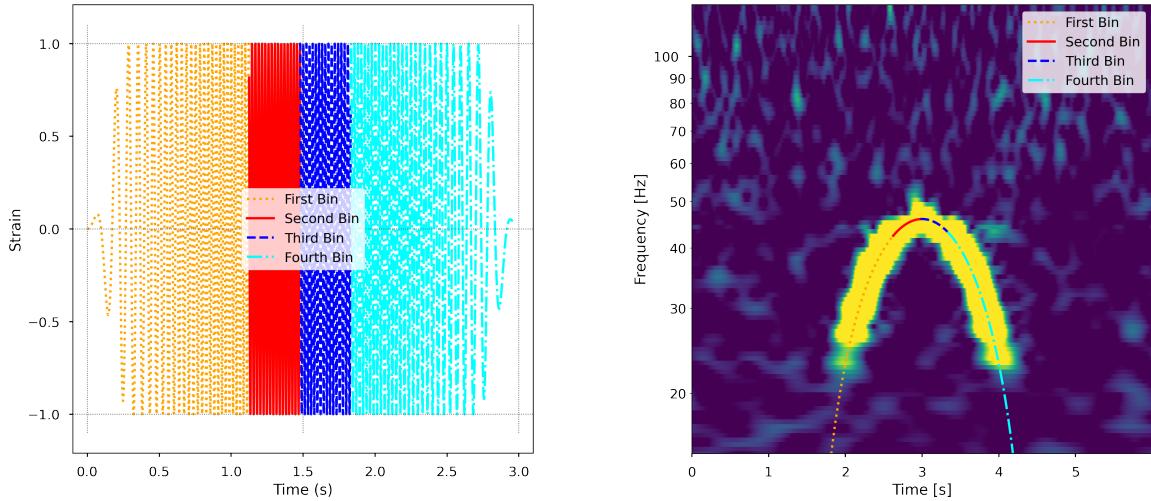


Figure 4.5: A scattered-light glitch template (left) where the colours and line-styles are indicative of the four equal SNR time bins to be used in calculating the  $\chi^2_r$  value and re-weighting the SNR. The same scattered-light glitch template bins overlaid on an injection of the scattered-light glitch (right). The inner two bins are considerably shorter than the outer two bins, which informs us that the centre—higher frequency—region of the scattered-light glitch contributes a larger amount to the SNR per unit time than the lower frequency regions.

one of the divided templates and the data. This test is constructed so as to produce large values when the data contains a glitch, or astrophysical signal, that is not well described by the template, but to follow a  $\chi^2$  distribution if a glitch that matches well to the template is present, or if the data is Gaussian and stationary.

The  $\chi^2$  test that we employ is similar to that of [?], however CBC waveforms increase in frequency over time whereas scattered-light glitch templates are symmetric about their centre. We therefore choose to construct our  $\chi^2$  test with four non-overlapping bins *in the time domain*, each of which contributes equally to the SNR, an example of the split template can be seen in figure ???. The matched filter between the bins and data is computed, and the  $\chi^2_r$  value is calculated using equation ?? (where  $n = 4$ ). Any glitch, or astrophysical signal, which does not exhibit symmetric morphology should not fit with this deconstruction of the template, and should result in elevated  $\chi^2_r$  values.

After computing the  $\chi^2_r$  value for potential scattered-light glitches, we follow [?] to compute a “re-weighted SNR”, which is an empirically tuned statistic depending on the SNR and the  $\chi^2_r$  value. The re-weighting function we use matches that presented



in [?],

$$\rho_{rw} = \begin{cases} \rho & \text{if } \chi_r^2 \leq 1, \\ \rho[(1 + (\chi_r^2)^3)/2]^{-\frac{1}{6}} & \text{if } \chi_r^2 \geq 1, \end{cases} \quad (4.16)$$

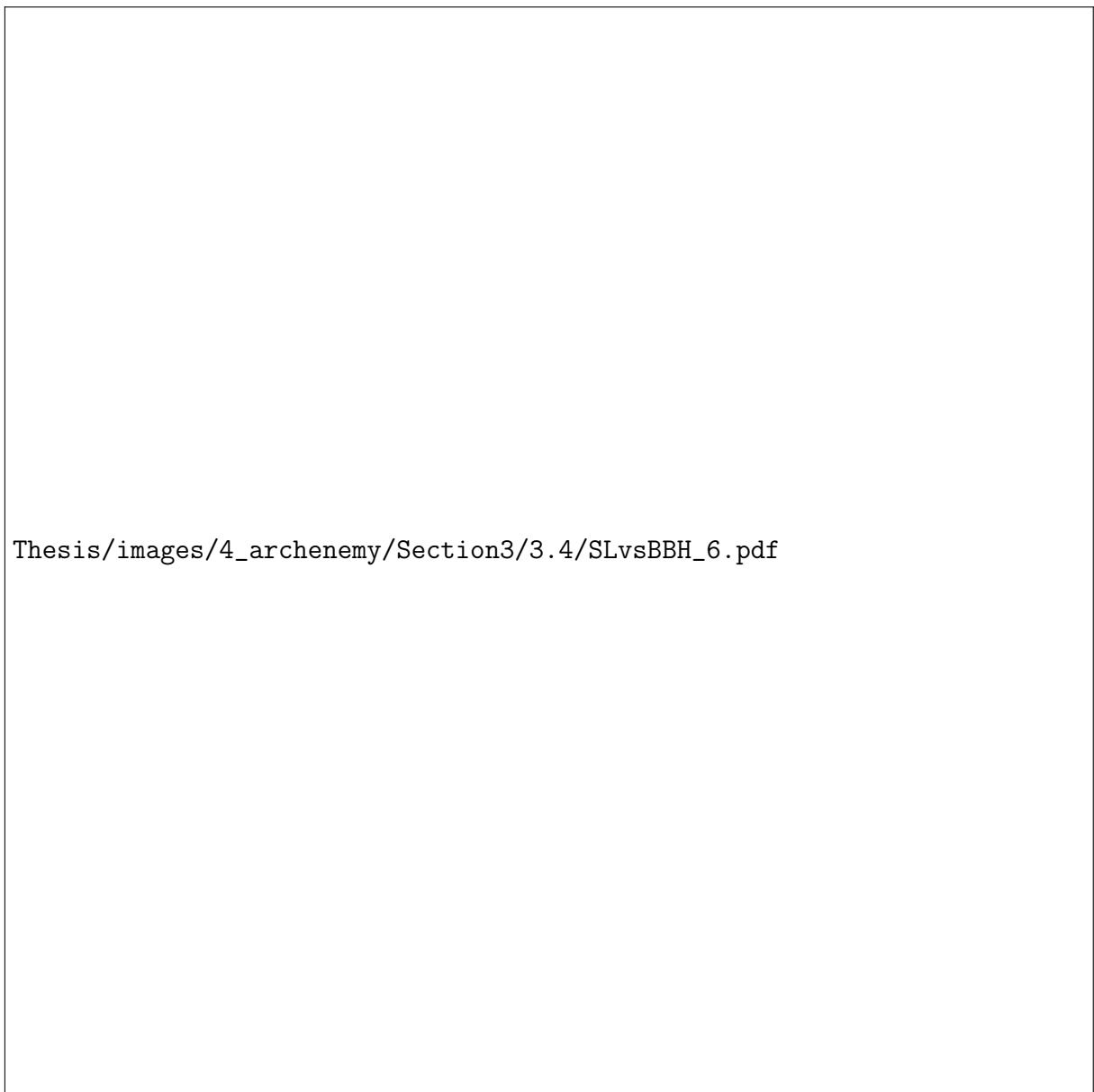
where  $\rho_{rw}$  represents the re-weighted SNR of the scattered-light glitch template calculated using the SNR,  $\rho$ , and the  $\chi_r^2$  value of that template. We do note that this re-weighting function has been tuned for CBC waveforms, and we did not repeat that tuning here with scattered-light glitches. One could retune this statistic, specifically targeting scattered-light glitches, using (for example) the automatic tuning procedure described in [?]. However, we demonstrate the suitability of the  $\chi^2$  test for our purposes in Figure ?? where we show the  $\chi_r^2$  vs SNR distribution of the triggers found by our scattered-light glitch search when performed on data containing only scattered-light glitches and data containing a binary black hole gravitational-wave injection.

We note that the  $\chi^2$  test increases the number of matched filters required by the search and therefore the computational cost of the search. Each template would require the matched filtering of an additional 4 “binned” templates to calculate the  $\chi_r^2$  value to re-weight the SNR time series of that template, increasing computational cost by a maximum factor of 5. However, we reduce this increase by only computing the  $\chi^2$  where it is needed, specifically for any template where the SNR time series has values above the threshold of 8.

### 4.3.5 Identifying all scattered-light glitches in the data

Our matched filtering process retains “triggers” (potential scattered-light glitches) wherever the re-weighted signal-to-noise time series is larger than 8. We retain no more than one trigger within a window size equal to half the *time period* of the template used to produce the re-weighted signal-to-noise time series, and only store triggers at local maxima. A re-weighted signal-to-noise time series with multiple peaks and identified triggers can be seen in the top right panel in figure ??.

After matched filtering all the templates against the data, we will recover multiple triggers for any potential scattered-light glitch, as we might expect to independently identify peaks in multiple templates around the true values of the glitch. We therefore collect all the triggers generated by the template bank and cluster these in time, using a window of half of the shortest duration template—0.9 seconds. This will result in a list of triggers corresponding to the highest re-weighted SNRs, where each trigger should correspond to a unique scattered-light glitch. The bottom left panel in figure ?? shows an example of the triggers found by the search and the highest re-weighted SNR triggers found by clustering.



Thesis/images/4\_archenemy/Section3/3.4/SLvsBBH\_6.pdf

Figure 4.6: The signal-to-noise ratio (SNR) and  $\chi^2$  values for the triggers identified by the matched filtering and clustering of the scattered-light glitch template bank with data containing only scattered-light glitches (grey hexagons) and data containing a binary black hole gravitational-wave injection (red triangles). The black contour lines represent the re-weighted SNR values the trigger will take when equation ?? is applied. The orange dashed vertical line indicates the SNR value limit of 8, above which we decide to perform the  $\chi^2$  test and calculate the re-weighted SNR. The blue solid contour line indicates a re-weighted SNR value of 8, which is the limit at which we consider the trigger to be real. Different re-weighting parameter values will produce different contour line shapes. It can be seen that no triggers for the data containing the gravitational-wave injection lie beneath the contour line, and therefore no scattered-light glitches are found on the gravitational-wave signal.

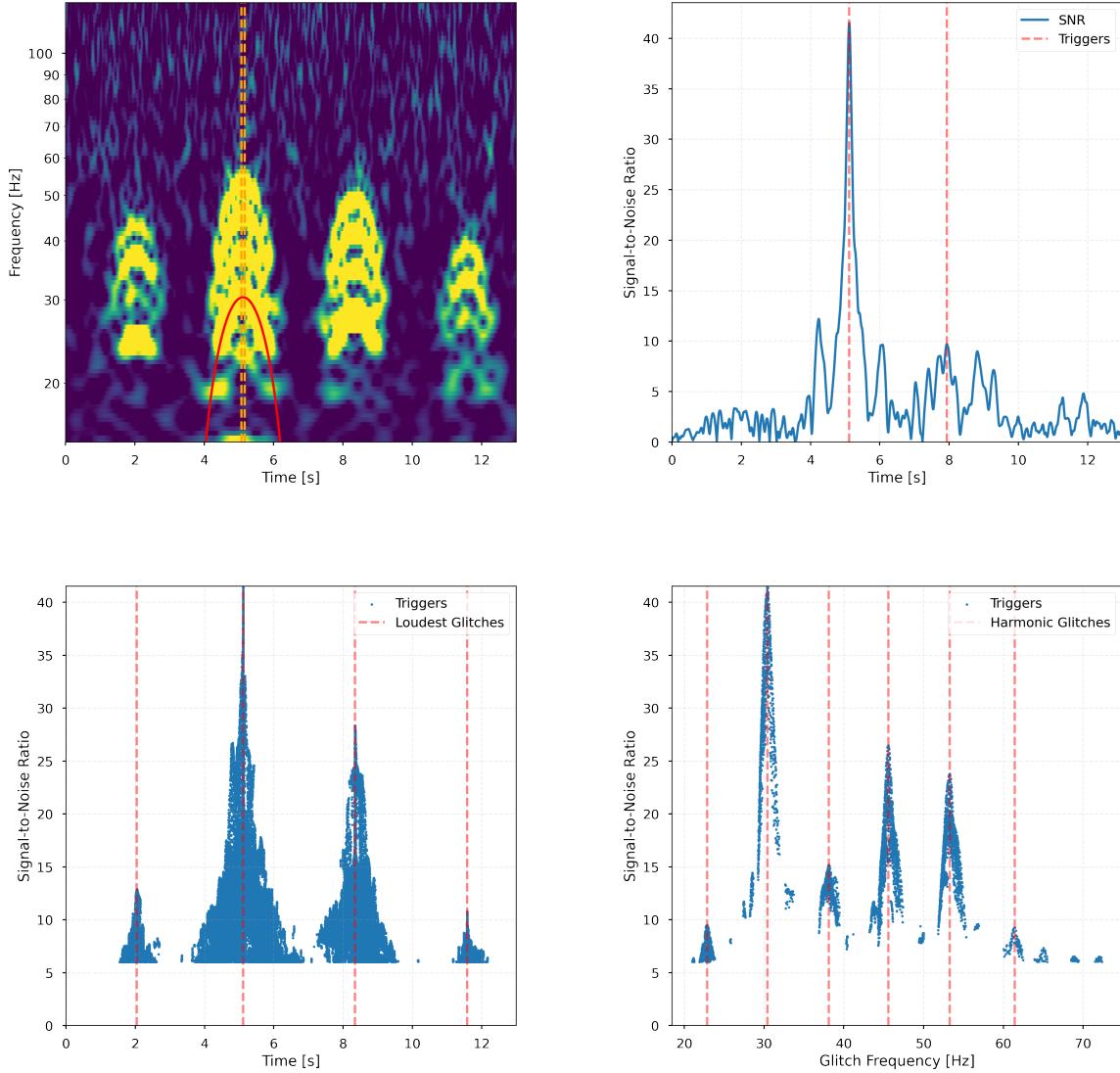


Figure 4.7: The process for identifying all the scattered-light glitches in a period of gravitational-wave data. The red overlay in the gravitational-wave data used in this example (top left) indicates the highest re-weighted signal-to-noise trigger found, the dashed vertical lines represent the time slice window around this trigger. The re-weighted SNR time series resulting from the matched filter of this trigger's template and the data (top right) is clustered in time to identify the triggers found above a signal-to-noise threshold of 8, indicated by red vertical dashed lines. All the triggers from all the templates in the template bank are then clustered in time (bottom left) to identify the highest re-weighted signal-to-noise glitches in the data, indicated by the orange vertical dashes lines. The triggers found within the time slice window, with a similar *time period* value—within  $\pm 10\%$ —of the highest re-weighted SNR trigger (bottom right) are clustered by their *glitch frequency* value to find the harmonic glitches at the same time, indicated again by red dashed lines.



However, we also expect to see instances of harmonic glitches which are produced by the same scattering surface and so share the same *time period* and have *glitch frequency* values equal to a multiple of the lowest frequency glitch in the harmonic stack [? ]. We investigate each trigger in the list, searching for harmonic glitches occurring at the same time. We use the first list of triggers identified by all templates across the bank and filter by those that occur within  $\pm 0.05$  seconds of the *centre time* of the trigger we are investigating, an example of this window can be seen in the top left panel of figure ???. We then filter the triggers again, keeping only those with an associated *time period* value within  $\pm 10\%$  of the trigger's *time period*. Finally, we cluster these remaining triggers by their associated *glitch frequency* using a window size of 4Hz, a lower limit on the frequency separation of harmonic glitches, the bottom right panel in figure ?? shows the identification of harmonic glitches for the overlaid scattered-light glitch in the top left panel of figure ??.

#### 4.3.6 Hierarchical subtraction to find parameter values

We now have a list of identified scattered-light glitches and their parameter values, however, these might not be fully accurate when there are many glitches close in time and frequency, as illustrated in Figure ???. It is important that the parameters we find match well with the glitches in the data to remove as much power as possible.

To better identify the parameter values of the scattered-light glitches, we perform a hierarchical procedure using information about the glitches we have found so far. Firstly, we create new segments of time which we know contain scattered-light glitches, taking a window of 8 seconds on either side of each previously identified glitch, if two glitch windows overlap they are combined into the same segment.

For each segment, we then create a reduced template bank, consisting of templates “close” to the scattered-light glitches previously identified in the segment. We take the smallest and largest *time period* and *glitch frequency* glitches in the segment and bound the retained templates by these values with  $\pm 0.25$  seconds on the *time period* and  $\pm 1\text{Hz}$  on the *glitch frequency*. For each data segment the reduced template bank is matched filtered with the data, the maximum re-weighted SNR value is found, and the corresponding glitch is subtracted. We then matched filter *again* and remove the next largest re-weighted SNR template. This process is repeated until no templates, when matched filtered with the data, produce any re-weighted SNR values above the SNR limit of 8. This method of hierarchical subtraction produces our final list of scattered-light glitches.

A further benefit of using these shorter data segments is that we are estimating the



PSD of the data using only a short period of data close to the scattered-light glitches being removed. This protects us from a rapidly changing PSD in non-stationary data, which might cause Gaussian noise to be identified with larger SNR in the periods where the PSD is larger. This can be resolved by including the variation in the power spectral density as an additional statistic in the re-ranking of triggers, which has been done for CBC gravitational-wave searches in [? ].

We demonstrate the hierarchical subtraction step on an amount of data which contains four injected scattered-light glitches in a single harmonic stack, this can be seen in figure ???. As shown, the scattered-light glitches are found and subtracted from the data leaving behind a cleaned segment of gravitational-wave data with no excess noise. Figure ?? shows the identified scattered-light glitch triggers before and after performing the hierarchical subtraction step on a stretch of data containing real scattered-light glitches. We identify more triggers prior to performing hierarchical subtraction, however there are more errant mismatches between scattered-light glitches and the overlaid templates. By performing the hierarchical subtraction, we more accurately identify scattered-light glitches, however, we miss some glitches that were previously identified. There is still some imperfection in this process, and we do not refine the method further in this work, but highlight this as useful direction for future studies in removing scattered-light glitches.

### 4.3.7 Identified scattered-light glitches

The methodology described in previous sections is implemented in our “ArchEnemy” pipeline, which is capable of searching for scattered-light glitches in gravitational-wave data using a pre-generated bank of glitch templates. We use ArchEnemy to analyse the aforementioned data, which produced a list of 2749 scattered-light glitches in data from the LIGO-Hanford observatory and 1306 from the LIGO-Livingston observatory.

The number of scattered-light glitches found by the ArchEnemy pipeline can be compared to Gravity Spy for the same period of time. Gravity Spy finds 2731 and 1396 for LIGO-Hanford and LIGO-Livingston respectively [? ]. There will be a difference in the number of glitches found by ArchEnemy and Gravity Spy for at least two reasons: Gravity Spy treats an entire stack of harmonic glitches as a single scattered-light glitch, whereas ArchEnemy will identify each glitch as a separate occurrence. Gravity Spy can also identify scattered-light glitches which are not symmetric and fall outside our template bank, for example, the scattered-light glitches shown in Figure ??.

Figure ?? is an example of the results of the ArchEnemy pipeline and how well it

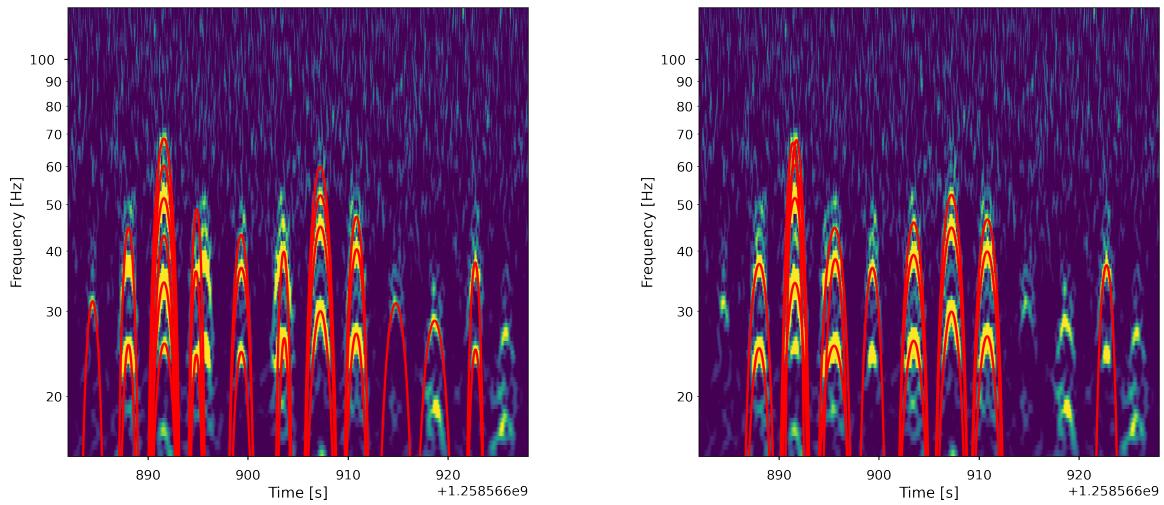


Figure 4.8: LIGO-Hanford data from 2019-11-23 17:54:22–2019-11-23 17:55:12 containing scattered-light glitches which have been identified by the ArchEnemy search (left), there is a misalignment in the template found for a number of glitches in this period of data and some missed glitches. Instead of finding all scattered-light glitches at once we run a hierarchical subtraction to find the scattered-light glitches in a chunk of time one-by-one. Scattered-light glitches remaining after running the hierarchical subtraction search (right) for the same period of data, we have missed more scattered-light glitches however misalignments have been removed. The highest harmonic at approximately 892 seconds has been incorrectly split into two separate templates.

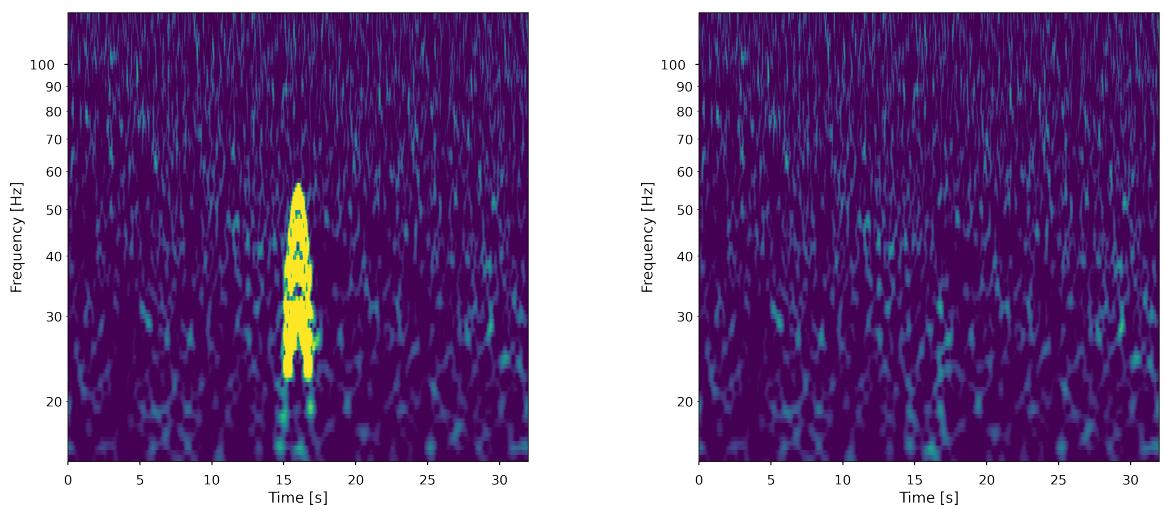


Figure 4.9: Data containing an injected stack of harmonic scattered-light glitches (left) and the corresponding data found when running the hierarchical subtraction search and subtracting the identified scattered-light glitches from the data (right).

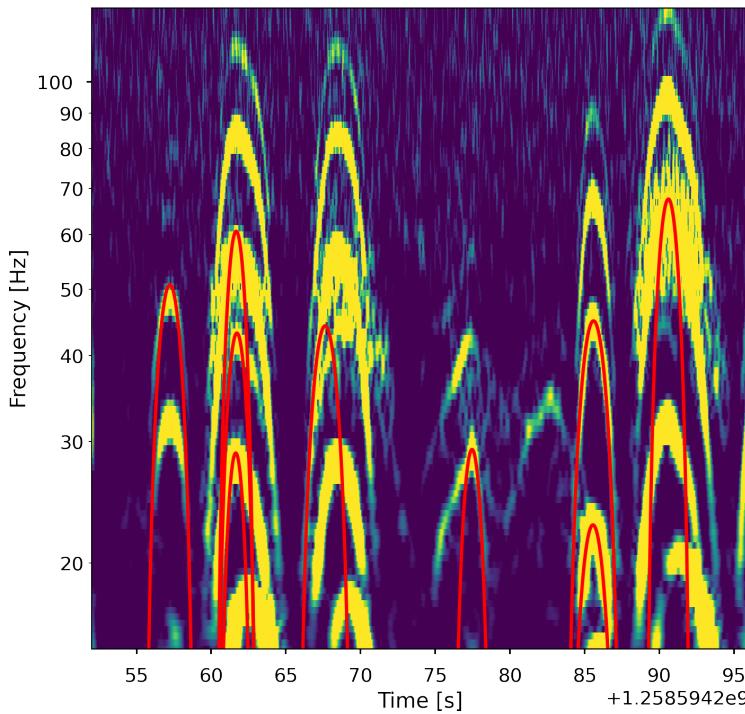


Figure 4.10: LIGO-Livingston data from 2019-11-24 01:30:32–2019-11-24 01:31:20 containing a very large number of scattered-light glitches at multiple times and frequencies over-plotted with the scattered-light glitches identified by the ArchEnemy search. Very few overlays match well onto scattered-light glitches. The template bank used in this search terminates at 80 Hz and so the scattered-light glitches located above this value will not be correctly identified. There are also asymmetric scattered-light glitches located which will not be identified correctly by our search, which assumes symmetry in the scattered-light glitch.

has identified scattered-light glitches in a period of data. A majority of the glitches have been identified with the correct parameter values and even in cases where the chosen template is not visually perfect, there is a good match between the template and the identified power in the data, particularly in the case of slightly asymmetric glitches. Figure ?? demonstrates a period of time when the ArchEnemy pipeline has not fitted well the scattered-light glitches in the data. The glitches at this time are improperly fit by the templates due to asymmetry of the morphology of the glitches and because some glitches are outside our template bank parameter range. However, we note that this is a very extreme period of scattered-light glitching and immediately after this time the detector data is no longer flagged as “suitable for analysis”.

We have demonstrated the ArchEnemy pipeline on a stretch of data from O3 and



have identified and characterised a list of scattered-light glitches, which could be removed from the data. We do note that there are cases where the identification has not worked well, but we expect that subtracting our list of glitches from the data will reduce their effect on the gravitational-wave search. In the next section, we will demonstrate this by quantifying sensitivity with the PyCBC pipeline.

### 4.3.8 Safety of scattered light identification

The data we have searched through contains no previously identified gravitational-wave signals [? ]. However, there is a risk that the ArchEnemy search would identify real gravitational-wave signals as scattered-light glitches. To assess this possibility, we simulate and add many gravitational-wave signals into the data and assess whether any signals are misidentified.

To do this, we use three separate sets of simulated gravitational-wave signals (or “injection sets”), one for BBHs, another for binary neutron stars (BNSs) and a third for neutron star black hole (NSBH) systems. We use the same simulations as the LVK search of this data, detailed in the appendix of [? ]. Each injection set consists of 6200 simulated signals spaced between 82 and 120 seconds apart. We treat these injection sets exactly the same as for the injection-less data, adding the simulations to the data, and then running ArchEnemy to produce a list of scattered-light glitches for each injection set.

To determine whether we have misidentified any gravitational-wave injections as scattered-light glitches we look for glitches we have found within the overlapping frequency band of gravitational-wave signals and our scattered-light glitch template bank. This corresponds to approximately 15 seconds before merger time for the injections. The simulated signals occur every  $\sim$ 100 seconds so we expect to see glitches within this 15 second window, therefore, we also require that there must be more triggers identified in the scattered-light glitch search *with* injections when compared to the search *without* injections within the window. The details of the number of gravitational-wave injections with overlapping scattered-light triggers can be seen in Table ??.

A scattered-light glitch will be identified close to a gravitational-wave signal in two cases: the ArchEnemy search is misidentifying the gravitational-wave signal as a glitch *or* the simulated signal was added close to actual glitches and a change in the data has meant a different number of glitches has been identified. The presence of real scattered-light glitches means we might miss a gravitational-wave signal, therefore, we *do* want to find and subtract glitches close to gravitational-wave signals, but we do



Table 4.1: For both interferometers and all 3 injection sets we identify the number of injections which are found to have scattered-light glitches identified within 15 seconds of merger time (“Injections with Coincident Triggers”), along with the number of scattered-light glitches found within this window for these injections (“Scattered-Light Coincident Triggers”). We investigated each of these injections and recorded the number which actually had scattered-light glitches identified due to the injected gravitational-wave signal (“Actual Overlapped Injections”).

Interferometer	Injection Set	Injections with Coincident Triggers	Scattered-Light Coincident Triggers	Actual Overlapped Injections
H1	BBH	20	45	1
	BNS	23	50	2
	NSBH	38	73	7
L1	BBH	13	21	2
	BNS	18	30	2
	NSBH	35	56	16

not want to subtract power from the gravitational-wave signal itself. The scattered-light glitch  $\chi^2$  test was designed to prevent the matching of scattered-light glitch templates to other causes of excess power, however, these results show it is not perfect.

We investigate each injection with coincident scattered-light triggers, seeing how many had misidentified scattered-light glitches on the inspiral of the gravitational-wave signal, this number can be seen in the column “Actual Overlapped Injections” in Table ???. We have included an example of the matching of scattered-light glitches onto gravitational-wave injections in Figure ??, the right panel shows the gravitational-wave data post glitch subtraction where it can be seen there is a portion of the power being subtracted from the signal. Although power is being removed from the signal, the gravitational-wave injection is still found by the search for gravitational waves, which we will describe later. For the cases that we have investigated, we note that the behaviour shown in Figure ?? only happens for signals that have very large SNR, and are therefore unphysically close to us. A similar effect is observed with the “autogating” process, described in [? ], which prevents the detection of these loud signals. In contrast to the “autogating” though, signals like that illustrated in Figure ?? are still identified as gravitational-wave signals by the PyCBC search after scattered-light glitch removal.

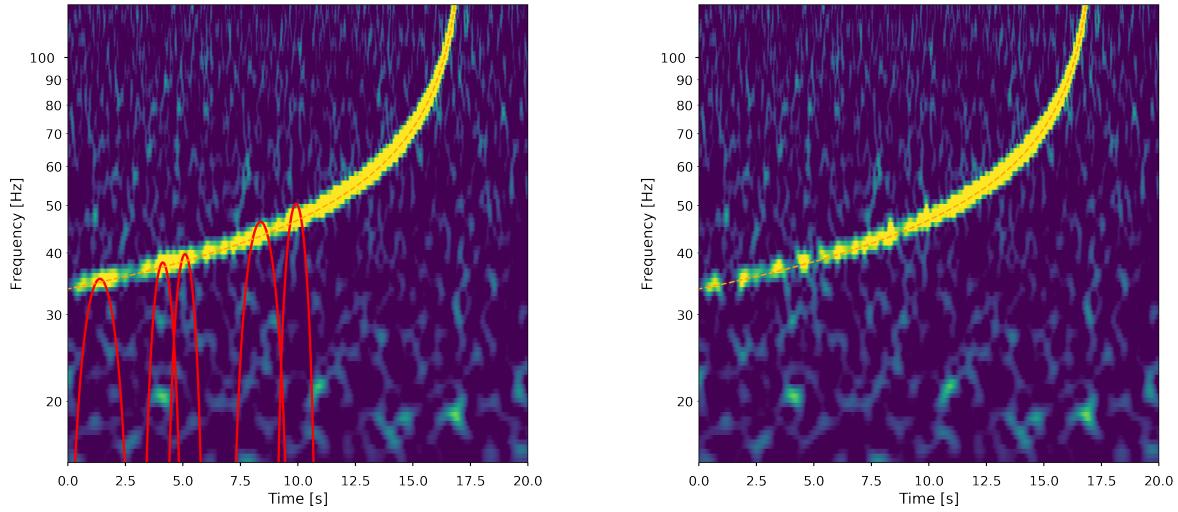


Figure 4.11: An injected binary neutron star compact binary coalescence gravitational-wave signal, with the scattered-light glitches identified by the ArchEnemy search pipeline overlaid in red (left). The same injected signal, but with the scattered-light glitches removed from the data (right). It can be seen that power is removed from the signal track and also there is an amount of power added to the data above the track.

## 4.4 Assessing sensitivity gain from removing scattered-light glitches

We now assess whether removing our identified list of scattered-light glitches results in a sensitivity gain when searching for CBCs. We do this by comparing the results from the offline PyCBC search on the original data, to the results of the same search but analysing data where the glitches have been removed.

### 4.4.1 Comparing search results with and without glitch subtraction

The PyCBC pipeline is able to assess the significance of potential CBCs in a given stretch of data, and does the same with a set of simulated signals. This significance is quoted in terms of a “false-alarm rate”, which denotes how often we would expect to see a non-astrophysical event at least as significant as the coincident trigger being considered. In this work, we assess sensitivity at a false-alarm threshold of 2 background events every year.

The data we have searched over contained no previously found gravitational-wave



Table 4.2: The number of injections found by each search with a false-alarm rate less than 2 per year alongside the number of newly-found and newly-missed injections, those found by the glitch-subtracted and not the original search and vice versa. We also show the sensitivity ratio of the glitch-subtracted search and original search for each injection set.

Injection Type	Original Search	Glitch-Subtracted	Sensitivity ratio	Newly Found	Newly Missed
BBH	1215	1222	1.01	10	3
BNS	1315	1315	1.00	5	5
NSBH	1260	1265	1.00	8	3

signals [? ] and our search after subtracting scattered-light glitches identified no new gravitational-wave signals. While the search hasn't found any gravitational waves, we can still measure the improvement in the sensitivity of the detectors by comparing the number of simulated signals identified with a false-alarm rate below 2 per year for each injection set (described in Section ??) with and without removing glitches from the data. Table ?? shows the number of injections found for all injection sets and both searches.

We compare the number of injections found by both searches but also look at the gravitational-wave injections found by the original search and missed by the glitch-subtracted search and vice versa, this information can be seen in Table ???. Considering signals found by the original search and missed by the glitch-subtracted search there are 3 binary black hole injections with false-alarm rates in the original search ranging from 0.5–0.3 per year, one of which had a glitch removed approximately 9 seconds after the injection, there are 5 newly-missed binary neutron star injections with false-alarm rates ranging from 0.5–0.056 per year, two of the five binary neutron star injections had glitches removed within 60 seconds of the injection, and there are 3 newly-missed neutron star black hole injections with false-alarm rates ranging from 0.5–0.086 per year, one had glitches removed within 60 seconds of the injection. The other newly-missed injections showed no scattered-light glitches within a 20 second window for binary black hole injections and a 60 second window for binary neutron star and neutron star black hole injections.

The glitch-subtracted search identifies 10 additional binary black hole injections, the most significant of which have false-alarm rates of 1 per 190.50, 1 per 7633.84 and 1 per 8643.73 years. We illustrate the last of these in Figure ?? (top). 5 extra binary neutron star injections were found, with false-alarm rates from 0.5–0.19 per year and 8 neutron star black hole injections were found, where the false-alarm rate of the most significant is 1 per 9961.55 years. This injection can also be seen in Figure ?? (bottom). We find 6 of the 10 binary black hole injections have scattered-light glitches within a 20



second window of the injection, 3 of 5 binary neutron star injections have scattered-light glitches within a 60 second window of the injection and, 6 of the 8 neutron star black hole injections have scattered-light glitches within a 60 second window of the injection. We provide more details about the newly-found and newly-missed injections in ??

To quantify the sensitivity of the search, we calculate the sensitive volume in which we can observe gravitational-wave signals. To calculate the sensitive volume we measure the detection efficiency of different distance bins taken from the injection sets and then multiply the efficiencies by the volume enclosed by the distance bins, these volumes are then summed to find the total volume the search is sensitive to [? ]. We are then able to calculate the ratio in sensitivities between the glitch-subtracted gravitational-wave search and the original gravitational-wave search, revealing the improvement that subtracting scattered-light glitches has made.

Figure ?? displays the ratio of the sensitive volume measured for the glitch-subtracted gravitational-wave search and the original PyCBC gravitational-wave search, across different false-alarm rate values, we quote our sensitivity ratios at a false-alarm rate value of 2 per year. The same set of injected signals was used for both gravitational-wave searches, and therefore a direct comparison of search sensitivities can be made via this ratio. Disappointingly, the measured sensitivity improvement is small in the results we obtain. For the binary black hole injections we measure a sensitivity ratio at a 2 per year false-alarm rate of 1.01, for binary neutron stars 1.00 and neutron-star-black-holes 1.00.

The statistical significance of the sensitivity increase we report for the binary black hole injection set can be found by investigating the null hypothesis of seeing the same 1% increase under the assumption that the subtraction of scattered-light glitches does not actually increase sensitivity. When performing this analysis we find that our result is not statistically significant at the 95% confidence interval –i.e. there is a 5.24% chance that we would measure such an increase in sensitivity at least as large as this under the null hypothesis– (see ?? for details). However, the marginal sensitivity increase would not justify repeating the scattered-light glitch search and glitch-subtracted gravitational-wave search on a larger injection set, instead more work is needed to better identify and remove scattered-light glitches while remaining safe in the presence of gravitational-wave signals.

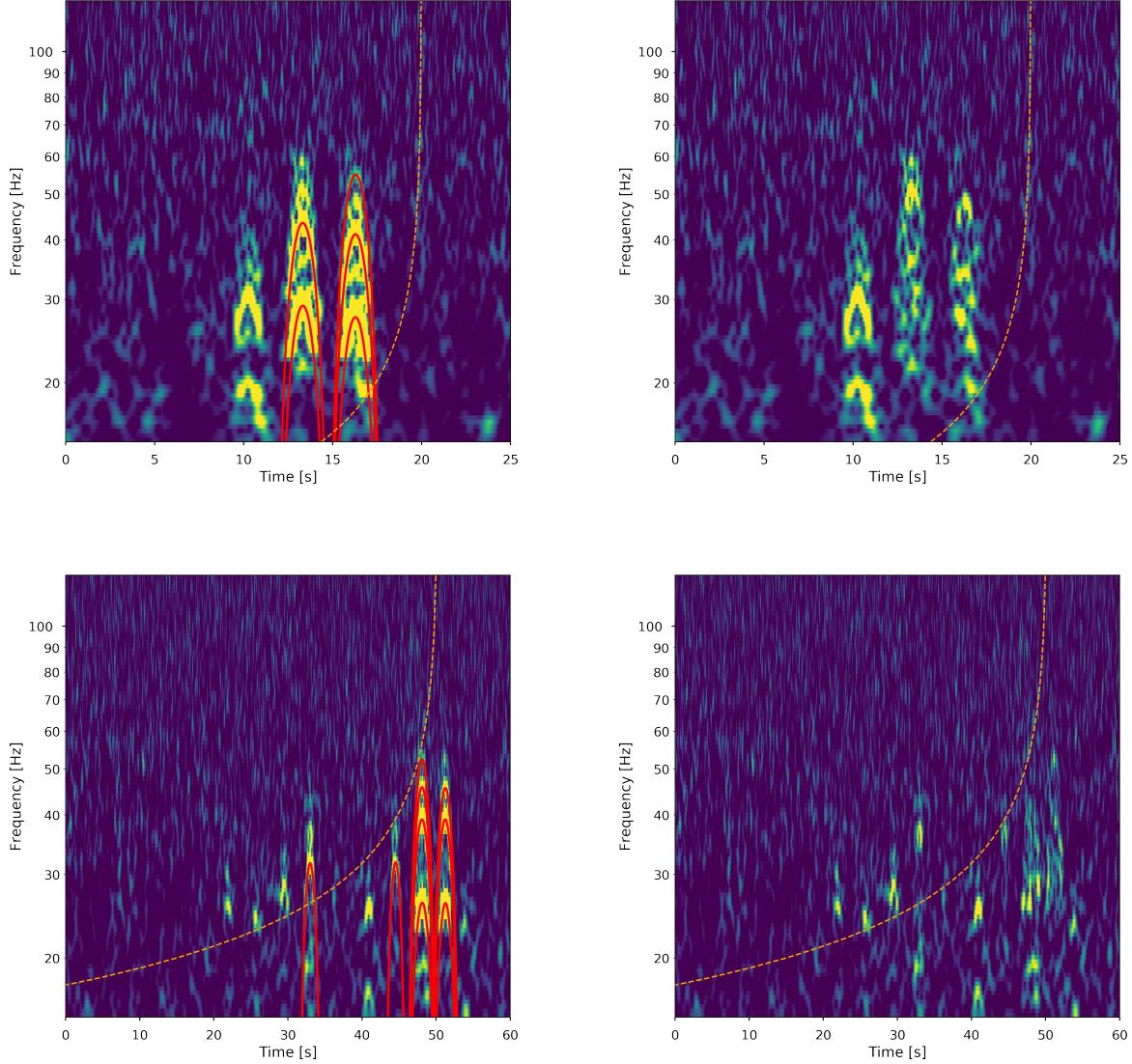


Figure 4.12: Two examples of gravitational-wave injections found by the glitch-subtracted search for gravitational waves which were not found by the original gravitational-wave search due to the presence of scattered-light glitches at the same time as the gravitational-wave inspiral. Top left: A binary black hole injection with a false-alarm rate of 1 per 8643.73 years is shown alongside the scattered-light glitches found by the ArchEnemy search and subtracted from the data prior to performing the glitch-subtracted PyCBC search for gravitational waves (top right). Bottom left: A neutron star black hole injection with a false-alarm rate of 1 per 9961.55 years and the scattered-light glitches found by the ArchEnemy search and subtracted from the data prior to performing the glitch-subtracted PyCBC search for gravitational waves (bottom right).

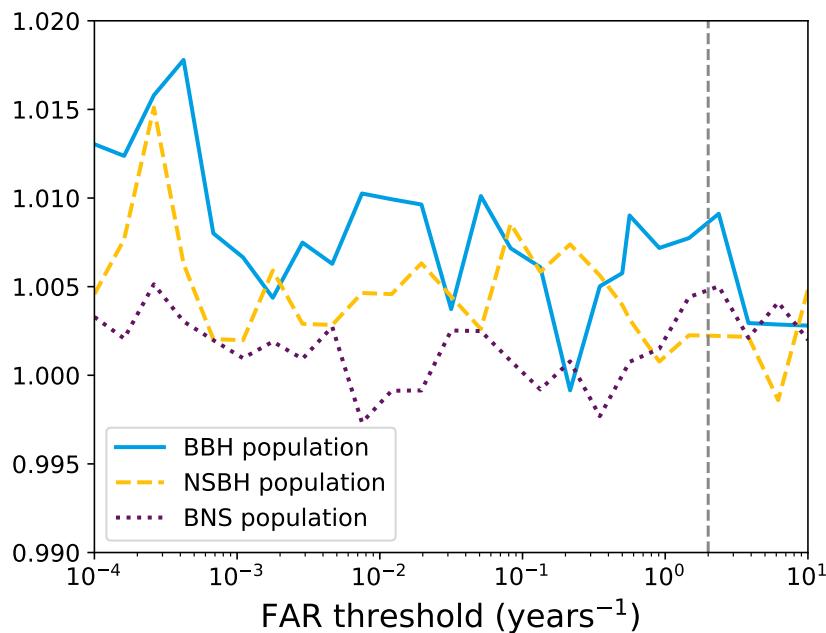


Figure 4.13: The ratio of the sensitive volume-time of the glitch-subtracted search and the original gravitational-wave search. The grey dashed line indicates a false-alarm rate of 2 per year, which is our threshold and the point at which we measure any sensitivity improvements of the glitch-subtracted search for each of the three gravitational-wave injection sets.

#### 4.4.2 Injections tables

Here we have three tables which contain data on the newly-found and newly-missed injections for each injection set. The tables are separated into the values for the inverse false-alarm rate and ranking statistic found for each injection in both searches then the SNR,  $\chi^2$  and PSD variation values for each detector in both searches. The first table is the results of the binary black hole injection set, the second table is the binary neutron star injection set and the third is the neutron star black hole injection set. A horizontal dashed line separates newly-found and newly-missed injections, using a false-alarm rate threshold of 2 per year. There were 2 newly-found binary black hole injections which were not found at all by the original search, and therefore they do not appear in the first table.

There have been 34 newly-found or newly-missed gravitational-wave injections when subtracting scattered-light glitches, it is informative to understand how the gravitational-wave search is influenced by the glitch subtraction to cause this outcome. The ranking statistic [? ] represents the legitimacy of a signal being astrophysical in origin and is partially computed using the re-weighted SNR, which is itself computed using the initial SNR alongside the various gravitational-wave discrimina-



tors and the PSD variation measurement [? ]. We use the trigger information saved by the gravitational-wave search to identify why injections that weren't found previously have been found post-glitch subtraction, and vice versa.

As an example, we take the smallest false-alarm rate (1 per 8643.73 years), newly-found, binary black hole injection, and look at the ranking statistic, SNR,  $\chi^2$  and, PSD variation measurements in both detectors and both searches – these values can be found in Table ???. This injection was originally seen with a false-alarm rate of 100 per year, far above our threshold, a very large increase in the ranking statistic from 13.29 to 27.01 is certainly responsible for the decreased false-alarm rate. There were no changes in the values measured by the LIGO-Livingston detector between searches which is expected as no scattered-light glitches were found within 512 seconds of the injection. The LIGO-Hanford detector sees a small increase in the SNR measured, from 7.98 to 8.22, a small increase in the  $\chi^2$  value, from 2.32 to 2.48, and a very significant decrease in the PSD variation measurement, from 3.47 to 1.50. Using equation 18 of [? ], we can calculate a re-weighted SNR of 4.60 for the original search and 5.67 for the glitch-subtracted search, a significant increase in the SNR. Similar analyses for all the newly-found and newly-missed injections can be made using information found in tables ?? and ??.

The decrease in the PSD variation is true for the three newly-found very low false-alarm rate injections, accompanied by the small changes in SNR and increase in the  $\chi^2$  measurement. For newly-found and newly-missed injections which lie close to the 2 per year false-alarm rate threshold, there is no definitive reason as to why these injections changed state.

Table 4.3: This table contains the trigger information for the newly-found and newly-missed **binary black hole injections** recorded by the original search [?] and the glitch-subtracted search.

		Glitch-Subtracted						Original Search							
IFAR	Ranking Stat.	H1			L1			IFAR	Ranking Stat.	H1			L1		
		SNR	$\chi^2$	PSD Var.	SNR	$\chi^2$	PSD Var.			SNR	$\chi^2$	PSD Var.	SNR	$\chi^2$	PSD Var.
8643.73	27.01	8.22	2.48	1.50	8.05	2.07	1.00	0.01	13.29	7.98	2.32	3.47	8.05	2.07	1.00
7633.84	26.13	8.18	2.25	1.22	7.58	1.46	1.03	0.37	16.63	8.19	1.87	2.17	7.59	1.45	1.02
4.89	19.06	7.23	1.88	0.97	10.34	3.15	1.05	1.56	18.01	7.24	2.11	0.97	10.34	3.42	1.04
3.14	18.65	5.89	1.54	1.01	8.86	2.40	1.09	1.52	17.99	5.89	1.54	1.01	8.86	2.40	1.09
2.75	18.53	8.50	1.63	0.99	6.80	1.37	1.05	1.51	17.98	8.51	1.56	0.98	6.80	1.37	1.05
2.39	18.41	7.83	2.00	1.01	8.41	3.41	1.02	1.06	17.65	7.81	2.14	1.01	8.41	3.41	1.02
2.18	18.32	7.02	1.90	1.02	6.35	2.15	1.01	1.79	18.14	7.05	1.92	1.02	6.35	2.16	1.01
2.01	14.11	7.94	1.89	1.01	5.67	1.95	1.16	1.90	14.14	7.94	1.90	1.01	5.07	0.00	1.02
1.76	18.12	6.89	1.43	1.02	6.76	2.59	0.97	3.33	18.71	6.86	1.82	1.02	6.68	2.53	0.97
1.79	18.14	6.40	1.69	1.02	7.56	2.01	1.01	2.05	18.27	6.40	1.69	1.02	7.56	2.00	1.01
1.65	18.06	5.14	0.00	0.91	7.57	1.57	1.02	2.05	18.27	5.14	0.00	0.91	7.57	1.57	1.02



Table 4.4: This table contains the trigger information for the newly-found and newly-missed **binary neutron star injections** recorded by the original search [?] and the glitch-subtracted search.

Glitch-Subtracted								Original Search							
		H1			L1					H1			L1		
IFAR	Ranking Stat.	SNR	$\chi^2$	PSD Var.	SNR	$\chi^2$	PSD Var.	IFAR	Ranking Stat.	SNR	$\chi^2$	PSD Var.	SNR	$\chi^2$	PSD Var.
5.12	19.12	5.86	1.99	1.01	7.15	1.94	1.02	1.49	17.96	5.74	2.14	1.01	7.15	1.94	1.02
4.91	19.07	5.48	2.29	1.01	7.92	2.16	1.01	1.37	17.89	5.78	2.25	1.01	7.23	2.13	1.01
4.54	18.99	4.82	0.00	1.07	8.68	2.05	1.06	1.66	18.07	4.82	0.00	1.14	8.68	2.05	1.06
2.46	18.43	5.48	1.95	1.05	7.61	2.03	0.99	1.78	18.13	5.48	1.94	1.05	7.62	2.08	0.99
2.34	18.39	6.25	2.18	1.01	6.82	1.96	1.00	1.45	17.94	6.09	2.05	1.01	6.83	2.01	1.00
1.24	17.71	6.58	2.14	1.01	6.89	2.48	1.01	17.81	20.19	6.58	2.15	1.01	7.16	2.26	1.01
1.14	17.72	6.00	2.67	0.98	8.07	2.66	1.22	7.14	19.44	6.17	2.45	0.97	8.26	2.80	1.22
1.89	18.19	6.32	2.19	1.00	6.79	1.83	0.99	3.57	18.78	6.37	2.12	0.99	6.79	1.75	0.99
0.88	17.46	6.88	2.14	0.99	6.19	2.12	0.99	2.33	18.39	6.87	2.07	0.99	6.19	1.99	0.99
1.36	17.80	6.20	2.02	1.00	6.83	2.35	1.00	2.07	18.18	6.20	2.02	1.00	6.83	2.28	1.00

Table 4.5: This table contains the trigger information for the newly-found and newly-missed **neutron star black hole injections** recorded by the original search [?] and the glitch-subtracted search.

		Glitch-Subtracted						Original Search							
IFAR	Ranking Stat.	H1			L1			IFAR	Ranking Stat.	H1			L1		
		SNR	$\chi^2$	PSD Var.	SNR	$\chi^2$	PSD Var.			SNR	$\chi^2$	PSD Var.	SNR	$\chi^2$	PSD Var.
9961.55	27.41	5.55	2.08	1.23	10.26	2.27	1.12	0.01	12.83	6.01	2.39	2.25	8.54	2.37	1.12
141.35	22.28	7.76	2.20	1.02	6.54	2.35	1.01	1.21	17.77	7.83	1.94	1.02	7.29	2.50	1.00
18.26	20.36	5.94	2.20	1.09	6.02	1.99	1.10	1.29	17.82	5.74	2.05	1.09	6.27	2.10	1.18
17.12	20.29	7.64	2.00	1.11	6.66	2.15	1.05	1.37	17.89	7.36	2.35	1.11	6.66	2.15	1.05
12.94	20.03	6.52	1.79	1.07	7.31	2.32	1.13	1.96	18.23	6.57	1.85	1.20	7.31	2.32	1.13
7.12	19.42	5.13	0.00	1.05	8.33	2.08	1.04	0.03	14.23	5.30	2.16	1.05	7.44	2.43	1.04
3.61	18.77	5.80	2.01	1.02	7.43	1.88	0.93	1.81	18.15	5.80	2.01	1.02	7.43	1.97	0.93
2.15	18.22	7.41	2.04	1.02	5.57	2.03	1.03	0.00	4.34	5.64	2.18	0.98	5.04	-0.00	1.01
0.23	16.16	5.79	1.81	0.96	8.78	3.93	1.03	11.69	19.93	5.78	1.89	0.96	8.79	3.17	1.03
1.56	17.92	6.71	1.91	0.99	6.73	2.50	1.05	2.68	18.41	6.71	1.90	0.99	6.72	2.39	1.05
1.72	18.10	5.77	1.90	0.99	6.77	1.74	1.00	2.08	18.28	5.77	1.90	0.99	6.80	1.69	1.00



## 4.5 Statistical significance

We report a 1% increase in the sensitivity for the binary black hole injection set in the glitch-subtracted gravitational-wave search (see Section ??). We wish to determine the statistical significance of this result under the null hypothesis that subtracting scattered-light glitches prior to searching for gravitational waves does not increase the sensitivity of the gravitational-wave search. The binary black hole injection set contains 6200 injected signals, 10 additional injections were found in the glitch-subtracted gravitational-wave search, 3 additional injections were missed in our search, this gives us 13 injections which have changed state.

First, we calculate the probability that an injection has been affected by the glitch removal,

$$p = \frac{13}{6200} = 0.21\%, \quad (4.17)$$

then we calculate the standard deviation,

$$\text{std} = \sqrt{n * p * (1 - p)} = 3.60. \quad (4.18)$$

Using the standard deviation we calculate the number of standard deviations our result deviates from the mean. We divide the number of positively changed (newly-found) injections by the standard deviation,

$$\text{standard deviation} = \frac{(10 - 3)}{3.60} = 1.94. \quad (4.19)$$

Under the assumption of no sensitivity increase caused by the subtraction of glitches, we measure our result of +7 newly-found gravitational-wave injections to lie 1.94 standard deviations from an expected value of 0 newly-found injections. The critical value of a 95% confidence interval, that is to say there is a 1 in 20 chance of our null hypothesis being true, is 1.96 meaning our result is within the 95% confidence interval. We can describe this as there being a 5.24% chance that our result is not caused by the subtraction of glitches but is instead caused by random chance. To reduce the error in the computed sensitivity ratio a larger injection set test would be required, this would need a large time and computational power investment which we do not believe is justified in the case of such a minor increase in the sensitivity.



## 4.6 Conclusion

We have demonstrated a new method for modelling scattered-light glitches and identifying and characterizing these glitches in a period of gravitational-wave data. We have developed a scattered-light glitch specific  $\chi^2$  test which can discriminate between scattered-light glitches, other types of glitches and gravitational-wave signals. We have searched through a representative stretch of gravitational-wave data known to contain scattered-light glitches, found thousands of these glitches and subtracted them from the gravitational-wave data prior to running a search for gravitational waves. The results of this search include a small increase in the measured sensitivity of the gravitational-wave search for binary black hole gravitational-wave signals, and modest change to sensitivity for binary neutron star and neutron star black hole gravitational-wave signals.

We highlight that the task of accurately identifying and parameterizing scattered-light glitches in the data is not a trivial one, especially where there are repeated, and harmonic, glitches present in the data. We have developed a new  $\chi^2$  test to reduce the number of false identifications of scattered-light glitches, but we do still see cases where we have misidentified other glitches, and even some loud gravitational-wave signals, as caused by scattered light, and cases where we do not correctly identify, or parameterize, actual scattered-light glitches. Improving this identification process would be important in improving the efficacy of this process.

The possibility of using this model of scattered-light glitches as a bespoke application to gravitational-wave signals which are known to have coincident scattered-light glitches has been explored and implemented into Bilby [? ] to perform a parameter estimation of scattered-light glitches and removing these to produce glitch-free data [? ]. The inclusion of the extra term from [? ] within the model can help identify scattered-light glitches more accurately. Selectively subtracting glitches based on the presence of a gravitational-wave signal is possible by moving the glitch subtraction process inside the gravitational-wave searches and including the results of other gravitational-wave discriminators [? ? ] to determine the legitimacy of an ArchEnemy identified glitch.

The results of the application of the ArchEnemy search pipeline, the list of scattered-light glitches, can also be used in other applications. For example, it could be used in the form of a veto [? ], where we use knowledge of the presence of scattered-light glitches to down rank periods of time in gravitational-wave data. Additionally, we could use scattered-light glitches previously identified by tools such as Gravity Spy [?] and target known scattered-light glitches with the ArchEnemy search pipeline.



As a final note, while we acknowledge that the sensitivity improvements that we have observed— $\sim 1\%$ —are very modest, the concept of removing scattered-light glitches, or other identified glitch classes, from the data prior to matched filtering for CBCs is one that we encourage others to explore further. An increase in the rate of events or the rate of scattered-light glitches in future observing runs will mean an increase in the number of affected events, such techniques offer a method for mitigating the effect that these glitches will have on the search, maximizing the number of observations that can be made.

# 5

---

## Improving the PyCBC Live Ranking Statistic

---

*“Have patience, all will be explained.”*

— Guthix, *The World Wakes*

This work will be published as part of the PyCBC Live fourth observing run methods paper, which is currently being drafted. I am responsible for the changes concerning the exponential noise model in the PyCBC Live ranking statistic as well as contributing to the development necessary to introduce these changes in the PyCBC Live infrastructure.

### 5.1 Introduction

Searching for gravitational waves is performed by search pipelines. The search pipelines operate across two timelines: low-latency, rapid detection of gravitational-wave signals in real time, and offline, which is performed months after the data has been obtained. Low-latency search pipelines are optimised for rapid detection to disseminate information about potential gravitational-wave signals to the wider scientific community in as little time as possible. The latency between a gravitational-wave signal arriving at the detectors and being detected by search pipelines is crucial for multi-messenger events.



Several offline gravitational-wave search pipelines were operational during the third observing run: cWB [? ], GstLAL [? ], MBTA [? ], and PyCBC Offline [? ]. Table XIV in Appendix D 7a of the second half of the third observing run’s catalogue paper [? ] highlights that the PyCBC Offline search was the most sensitive gravitational-wave search pipeline [? ? ? ].

The low-latency gravitational-wave search pipelines include cWB [? ], GstLAL [? ], MBTA [? ], PyCBC Live [? ], SPIIR [? ], and oLIB [? ]. In the third observing run, the GstLAL pipeline [? ] uploaded the preferred event<sup>1</sup> for 33 of the 56 low-latency super events, whereas PyCBC live uploaded the preferred event in only 15 events [? ]. The PyCBC live search pipeline is not the most sensitive in the low-latency regime, so we want to improve the PyCBC Live search to have the same sensitivity as its offline counterpart, and to do this we must look at the key differences between the two searches.

Offline searches for gravitational waves can utilise post-detection information when processing their results and have a much larger limit on the amount of computational analysis that can be performed to find events. Low-latency gravitational-wave searches, such as PyCBC Live, receive and analyse data in regular fixed analysis segments, and all computational processing must be completed before the next analysis segment arrives. If the triggers in an analysis segment are not processed in time, lag is introduced as a backlog of triggers builds up faster than they can be processed. The maximum latency between an event arriving and being detected by PyCBC Live during the third observing run was 20 seconds [? ].

PyCBC Offline contains components and information that could not be previously used by the PyCBC Live search without violating the latency restrictions. Introducing these components into the live search will improve the sensitivity of the gravitational-wave search, bringing the two PyCBC searches closer to parity and enabling the detection of more gravitational-wave events with greater significance.

This chapter is laid out as follows: In Section ?? we describe how PyCBC ranks identified events to assess significance, in Section ?? we discuss the ranking statistic used by the PyCBC searches during the third observing run and how these are constructed, in Section ?? we describe the new additions to the PyCBC Live ranking statistic we have implemented and how these have been adapted specifically for the PyCBC Live search, in Section ?? we discuss how we evaluate the sensitivity improvement of our changes using an injection study and in Section ?? we discuss the immediate sensitivity increase highlighted by the injection study. After this, we delve deeper into the regions of parameter space responsible for the sensitivity changes in

---

<sup>1</sup>The event with the highest SNR of those uploaded by all pipelines.



sections ?? and ?? . We describe the test of the ranking statistic changes on the PyCBC Live mock data challenge infrastructure in Section ?? and conclude in Section ??.

## 5.2 The ranking statistic

The ranking statistic is the detection statistic used to calculate the significance of a gravitational-wave detection. The ranking statistic values can be mapped directly to false-alarm rate (FAR), a key metric used to assess the likelihood that a detected signal is real and not a result of coincident noise triggers being found by the gravitational-wave search [? ].

A ranking statistic combines numerous pieces of information about a candidate event to provide a single measure that reflects the event’s significance. The ranking statistic can include information such as: single detector trigger signal-to-noise ratio (SNR),  $\rho$ , signal-consistency tests [? ? ? ] and, coincidence tests between detectors. It can also include more complex information like coincident phase and time difference likelihood based on source localisation and detector orientation [? ? ]. Different ranking statistics combine different pieces of information to make the final assessment of significance.

Improving the ranking statistic used in the PyCBC Live search will improve the search’s ability to distinguish between real gravitational-wave events and false-alarms. The PyCBC Offline search’s ranking statistic contains more information than the ranking statistic used by PyCBC Live during the third observing run, some of which we can adapt for the PyCBC Live search [? ? ? ].

## 5.3 PyCBC ranking statistics used in O3

The PyCBC Live search will calculate the single detector ranking statistic, the SNR re-weighted by the  $\chi^2$  tests, for each trigger for each online detector; coincidences are then formed between triggers to identify potential gravitational-wave events. The coincident triggers are then ranked by the coincident ranking statistic to provide the final ranking statistic value.

The PyCBC Offline search in the third observing run ranked single detector triggers by: SNR,  $\rho$ , calculated by the matched filter of template and data [? ]; the traditional  $\chi^2$ , calculated by measuring the difference in the expected and actual  $\rho$  for discrete frequency bins in the signal evolution [? ]; the sine-Gaussian  $\chi^2$ , which measures  $\rho$  in frequency bins above the signal’s maximum frequency [? ] and; PSD variation, which estimates the effect of non-stationary noise on  $\rho$ , re-weighting  $\rho$  prior



to applying the  $\chi^2$  tests [? ]. These four single detector ranking statistic components are applied to all triggers found by the search to give a new re-weighted  $\rho$  value,  $\hat{\rho}$  [? ]. [? ] provides a detailed review of the  $\chi^2$  tests currently being used in gravitational-wave searches.

After the calculation of  $\hat{\rho}$  for each single detector trigger, the different detector triggers are combined and a coincidence test is applied to identify coincident triggers. The coincidence test checks whether the arrival time of the gravitational-wave signal at the separate detectors is possible given the light-travel time of the gravitational-wave signal, for example, the two LIGO detectors are separated by a straight line through the Earth 3002 kilometres long. Given the speed of light, there is a 0.01 second maximum physical travel time for a signal detected by one detector to appear in the other detector. On top of this time window, we add an additional 0.002 seconds to account for timing errors between the two detectors, giving a total allowed time difference between trigger arrival times at different detectors of 0.012 seconds.

Triggers across detectors found within the time window and above the  $\hat{\rho}$  threshold (typically 4.5) can be considered coincident and are passed to the coincident ranking statistic for ranking. The PyCBC Offline coincident ranking statistic includes: likelihood of coincident trigger time and phase differences [? ], likelihood correction based on data artefacts found in correlated auxiliary channels which are insensitive to gravitational waves [? ? ], an exponential noise model which models noise distributions of templates separately and re-ranks single detector ranking statistic based on how frequently a template has historically triggered on noise with high  $\hat{\rho}$  [? ], and a template dependent factor which re-weights templates based on their astrophysical likelihood [? ].

The PyCBC Live ranking statistic used during the third observing run was comparatively simple. The single detector ranking statistic is the same as the PyCBC Offline single detector ranking statistic, except without the PSD variation statistic. PyCBC Live only ranks coincident triggers by the likelihood of their time and phase differences [? ].

### 5.3.1 Constructing the O3 PyCBC Live ranking statistic

We describe the ranking statistic used by PyCBC Live during the third observing run in more detail to identify the potential components that can be added from the PyCBC Offline ranking statistic. The construction of the ranking statistic used by PyCBC Live during the third observing run is described in [? ].

As described in Section ??, the general optimal ranking statistic is defined as the



difference of the log of the signal rate,  $p^S(\vec{\theta})$  and noise rate,  $p^N(\vec{\theta})$ ,

$$R = \log p^N(\vec{\theta}) - \log p^S(\vec{\theta}) \quad (5.1)$$

over the parameter space  $\vec{\theta} = (\hat{\rho}_H, \hat{\rho}_L, \chi_H^2, \chi_L^2, \delta\phi, \delta t, m_1, m_2, \vec{s}_1, \vec{s}_2)$ .

The construction of the ranking statistic used by PyCBC Live,

$$R^2 \propto 2[\log p^S(\vec{\theta}) - \log p^N(\vec{\theta})] + constant, \quad (5.2)$$

was chosen such that the ranking statistic of a two detector coincidence with both Gaussian and stationary noise <sup>2</sup> will have single detector noise rate,  $r_n$ , that falls off as a Gaussian,

$$r_{n,det} \propto \exp\left(-\frac{(\rho_d - \mu)^2}{2\sigma^2}\right), \quad (5.3)$$

and a combined detector noise rate that is the product of both detector noise rates,

$$p^N(\vec{\theta}) \propto \exp\left(-\frac{(\rho_H^2 + \rho_L^2)}{2}\right), \quad (5.4)$$

recovering the standard quadrature sum SNR statistic,

$$R^2 = -2 \log p^N(\vec{\theta}) = \rho_H^2 + \rho_L^2. \quad (5.5)$$

To expand on the quadrature sum ranking statistic, which only considers the noise rate, we can include information about the signal rate which takes the parameters  $(\rho_H, \rho_L, \delta t, \delta\phi)$ : the two detector SNRs, the difference in the time of arrival for the two detectors,  $\delta t = t_H - t_L$ , and the difference in phase of the gravitational waveforms,  $\delta\phi = \phi_H - \phi_L$ . These parameters, for a real signal, will depend on the source localisation and detector orientation and therefore, some combinations of parameters are more likely than others. The exact process for calculating  $p^S(\vec{\theta})$  is provided in detail in [?], for this chapter's purpose a look-up is made to a phase-time histogram which represents the most likely combination of the two parameters in the signal space. Including  $p^S(\vec{\theta})$  in the ranking statistic (Equation ??) yields,

$$R = \sqrt{\rho_H^2 + \rho_L^2 + 2 \log(p^S(\vec{\theta}))} \quad (5.6)$$

which is the final version of the original ranking statistic used by PyCBC Live in the

---

<sup>2</sup>The detector noise is **not** assumed to be Gaussian and stationary, but this ranking statistic is recovered if that were the case.



third observing run.

## 5.4 Improving the PyCBC Live ranking statistic

The ranking statistic used by PyCBC Live in the third observing run combines the noise rate contribution, which is the quadrature sum of detector  $\hat{\rho}$  in stationary, Gaussian noise, and the signal rate of the trigger. We know our detector noise is **not** stationary and Gaussian [? ] and therefore by including an accurate noise model in the ranking statistic we can improve the sensitivity of the PyCBC Live search.

There are two components of the PyCBC Offline ranking statistic that directly address these two deficiencies in our ranking statistic: PSD variation, which corrects the single detector  $\rho$  for non-stationary noise present in the data and; the exponential noise model, which models the noise distribution of each template by fitting an exponential to the noise falloff.

These changes dramatically improve our noise model for the data, providing a more accurate noise rate for use in calculating the ranking statistic and assigning significance to our gravitational-wave events. A better noise model allows the ranking statistic to better be able to distinguish between signal and noise events. There are other components of the PyCBC Offline ranking statistic that are in development in parallel to our work to be used in the PyCBC Live ranking statistic in the future: a kernel density estimation method to estimate probability density functions of signal rate for templates in the template bank, applying a weighting to the signal rate of a coincidence trigger depending on the signal search parameters and the template used to find the signal [? ]; iDQ uses data quality information from auxiliary channels to provide more information to the noise rate for triggers in PyCBC Live [? ]. All of these changes will improve PyCBC Live’s significance estimation in future observing runs.

### 5.4.1 Adapting PyCBC Live to compute generic offline detection statistic in low-latency

We must discuss the architecture of the PyCBC Live [? ] search before we can discuss how the changes to the ranking statistic have been made. Data from the gravitational-wave observatories is distributed to computer clusters, such as the LIGO Data Analysis System (LDAS) at Caltech in the United States of America [? ], which is made available to the PyCBC Live search pipeline on a number of cluster nodes. PyCBC Live has exclusive access to 151 cluster nodes for the explicit purpose of running the



PyCBC Live full-bandwidth search pipeline, 150 of these nodes are ‘worker’ nodes and 1 (rank 0) is the ‘control’ node.

The template bank is split equally across these worker nodes which are independent and produce their own set of single detector triggers. The triggers from all worker nodes are transferred to the control node which combines them into coincident events, calculates ranking statistic and false-alarm rate. PyCBC Live has strong limitations placed on it by the low-latency nature of the search pipeline, the computational processing of a data segment must be completed prior to the end of the next data segment lest data will build up waiting to be processed. These latency requirements limit additions that can be made to the search pipeline because they might either take too long to compute or will require the data to wait for extra information before the processing can be completed.

PSD variation requires the matched filtering of a generic CBC filter to calculate the variation time series. This matched filter is calculated by the control node because the PSD variation values needs to be distributed to all worker nodes for single detector ranking statistic calculation. This calculation introduces a negligible amount of latency on top of the  $\sim 700,000$  matched filters already being performed every data stride. Section ?? details the implementation of PSD variation in live.

The template fits require a large number of triggers for each template in order for the noise model to be representative of the true noise distribution. In PyCBC Offline the template fits are created using the triggers produced by the templates in the whole segment of data. The template fit statistics are then calculated and the triggers are then re-ranked depending on the resulting template fit statistics of the templates. We would not be able to do the same for PyCBC Live because we cannot create the template fit statistics every search segment, these files can take up to an hour to produce.

To incorporate the exponential noise model into the PyCBC Live ranking statistic, we begin by assuming that the noise distribution remains approximately constant over a period of at least one week. This assumption simplifies the task of modelling the noise, allowing us to use data from recent PyCBC Live triggers and capture the same statistics for each template. Specifically, we generate the template fits based on the previous week’s worth of PyCBC Live triggers, and use these fits to produce the necessary statistic files. These statistic files are then employed to re-rank new PyCBC Live triggers.

This approach offers a significant advantage in terms of efficiency, as it allows the creation of template fit statistic files to occur in parallel with the PyCBC Live search, avoiding any additional latency in processing gravitational-wave events. Furthermore,



this system is flexible enough to accommodate regular updates to the statistic files. While we currently update the statistic files on a weekly basis, a more dynamic approach could involve a rolling window, where the noise model is refreshed more frequently—for example, by using a sliding week-long window that is updated every day. Such a rolling update would allow for a more refined, real-time adjustment to changing noise conditions. These can be tuned in the future.

### 5.4.2 PSD variation

The power spectral density (PSD) describes how much power is present in the data at each frequency, essentially how the data power is distributed in the frequency domain. The power in each frequency band can change over time due to instrumental noise such as seismic noise, thermal noise and quantum noise and cannot be mitigated except through detector improvements. We call this noise non-stationary noise. The noise profile can change over the period of a search segment, meaning the PSD will be inaccurate at different times. Identifying triggers in the PyCBC searches is done by matched filtering the data with a signal template, described in Section ??, which depends on the PSD in the noise-weighted inner product.

A PSD which doesn't accurately describe the noise profile of the data will lead to a misestimation in the calculation of  $\rho$ . Measuring the actual PSD of the data is particularly challenging due to the presence of non-stationary and non-Gaussian noise therefore, the offline search empirically measures the PSD of the gravitational-wave data being analysed using Welch's method, however, instead of using a mean average of the overlapping PSDs a median average it used to remove the effect of short duration glitches in the data.

In PyCBC Offline, the estimated PSD is used for very long search segments (512 seconds) and while the effects of short duration glitches have been mitigated with the median average method we still suffer from the effects of non-stationary noise.

The mismatch between the estimated PSD,  $S_E$ , and actual PSD,  $S_A$ , of the data will have the effect of underestimating and overestimating the noise at different times in our search segment. We are able to track the sources of the non-stationary noise in some cases, for example, when the source of the non-stationary noise is a period of increased seismic activity this will be identified by the seismometer auxiliary channels and can then be subtracted from the gravitational-wave data. This isn't true for all noise however, some non-stationary noise has no immediately identifiable source and we are unable to make the subtraction.

Therefore, we must track the difference between  $S_E$  and  $S_A$  using the PSD variation



statistic [? ]. The PSD variation statistic models the relationship between  $S_E$  and  $S_A$  as,

$$S_A = \nu_s S_E, \quad (5.7)$$

where  $\nu_s$  is treated as a frequency independent parameter. The PSD variation is defined as the time series which tracks  $\nu_s$ , therefore, by computing  $\nu_s(t)$  we can calculate the difference between  $S_A$  and  $S_E$  for all times in our search data.

In the offline search  $\nu_s(t)$  is calculated using an approximate expression for a typical CBC template,  $|h(f)| \propto f^{-\frac{7}{6}}$ , and  $S_E$  to construct a filter,

$$F = \frac{|h(f)|}{S_E}, \quad (5.8)$$

which is band-passed between 20Hz and 480Hz, smoothed with a Hann window and combined with the data to produce an equation for the  $\nu_s(t)$ ,

$$\nu_s(t) \equiv N\langle\rho\rangle(t), \quad (5.9)$$

where  $N$  is a constant such that the expectation value of  $\nu_s$  in Gaussian noise is 1 and  $\langle\rho\rangle$  is the variance of  $\rho$ . This produces  $\nu_s(t)$  that can then be applied to the offline search as part of the single detector ranking statistic where triggers are assigned  $\nu_s$  at trigger time,  $t$ , which re-weights  $\rho$  prior to any signal consistency tests,

$$\rho_{scaled} = \frac{\rho}{\sqrt{\nu_s}}. \quad (5.10)$$

The live search maintains a data ring buffer of 512 seconds, rolling the newest eight seconds of data on as the older eight seconds are rolled off. The live search can be operating for potentially weeks at a time, during which the detector noise can change, this means the live search requires a dynamic PSD which can update during runtime. The initial PSD is estimated using Welch's method with a median average and is first created when enough data has been accumulated after starting the search. Every analysis segment (eight seconds) a new PSD,  $S_N$ , is estimated and compared to the current search PSD,  $S_C$ . The comparison is made by calculating the distance a binary neutron star system with equal  $1.4 M_\odot$  masses would need to be from the detectors to be observed with  $\rho = 8.0$ , this is called the BNS distance and it depends exclusively on the PSD. The BNS distance is calculated both  $S_C$  and  $S_N$  and if the BNS distance of  $S_N$  differs from the BNS distance of  $S_C$  by  $\pm 1\%$  then  $S_N$  replaces  $S_C$ . If the BNS distance of  $S_N$  is within the threshold then  $S_N$  is discarded and the search keeps using  $S_C$ .



The live search does not need to calculate  $\nu_s(t)$  for the entire data ring buffer, new triggers are only ever found in the latest analysis segment and therefore this is the only period of time for which  $\nu_s(t)$  needs to be calculated and  $\nu_s$  values distributed to the new triggers. To calculate the PSD variation values we need to track  $S_E$  for each detector and create a new filter (Equation ??) for each detector which is updated every time  $S_E$  is updated. The filters are then convolved with the latest eight seconds in the data ring buffer, the mean square of  $\nu_s(t)$  is calculated every 0.25 seconds to find outliers caused by short duration glitches. These outliers are then replaced with an average of the adjacent elements in  $\nu_s(t)$  and the time series is then further averaged every second to produce the final  $\nu_s(t)$ . For each trigger,  $\nu_s$  is then extracted by interpolating between the two nearest whole seconds in the time series and is then used to scale  $\rho$  before  $\hat{\rho}$  is calculated, as shown in Equation ??.

### 5.4.3 Modelling the noise distribution in live

Non-Gaussian noise artefacts (glitches) can produce high SNR triggers, these are partially mitigated with the single detector ranking statistic signal consistency tests, but there is still a long tail in the noise distribution (Figure ??) that can be attributed to the effects of glitches. This noise falloff has historically been assumed to be Gaussian in the PyCBC Live search and all templates are treated equally. There are a number of very common glitches that occur frequently in the data that can match more closely to some templates in the template bank and not others. An example of this is blip glitches [? ] which resemble the gravitational-wave signals of high mass binary black hole mergers, therefore the templates which describe these signals will trigger more regularly on these glitches with higher  $\hat{\rho}$  compared to other templates in the bank which do not look similar to any glitch class. Not only will the noise falloff no longer be Gaussian, each template is treated equally, where some trigger far more frequently with higher  $\hat{\rho}$ .

Single detector triggers found by PyCBC searches are only saved if the  $\rho > 4.5$ . All triggers above the threshold from the noise distribution we want to model. The noise falloff is modelled in PyCBC by taking all the triggers produced by a template and fitting an exponential function to the slope of the falloff giving the exponential fit factor,  $\alpha$ , and the count of triggers found with  $\hat{\rho} > 6.0$ ,  $\mu$ . This process is commonly called ‘template fitting’ and the output parameters are called the ‘template fits’.

PyCBC Offline searches through pre-defined chunks of IGWN data [? ], which are all around one calendar week long. The search will matched filter the template bank and the data in one of these chunks, and all triggers above the previously mentioned

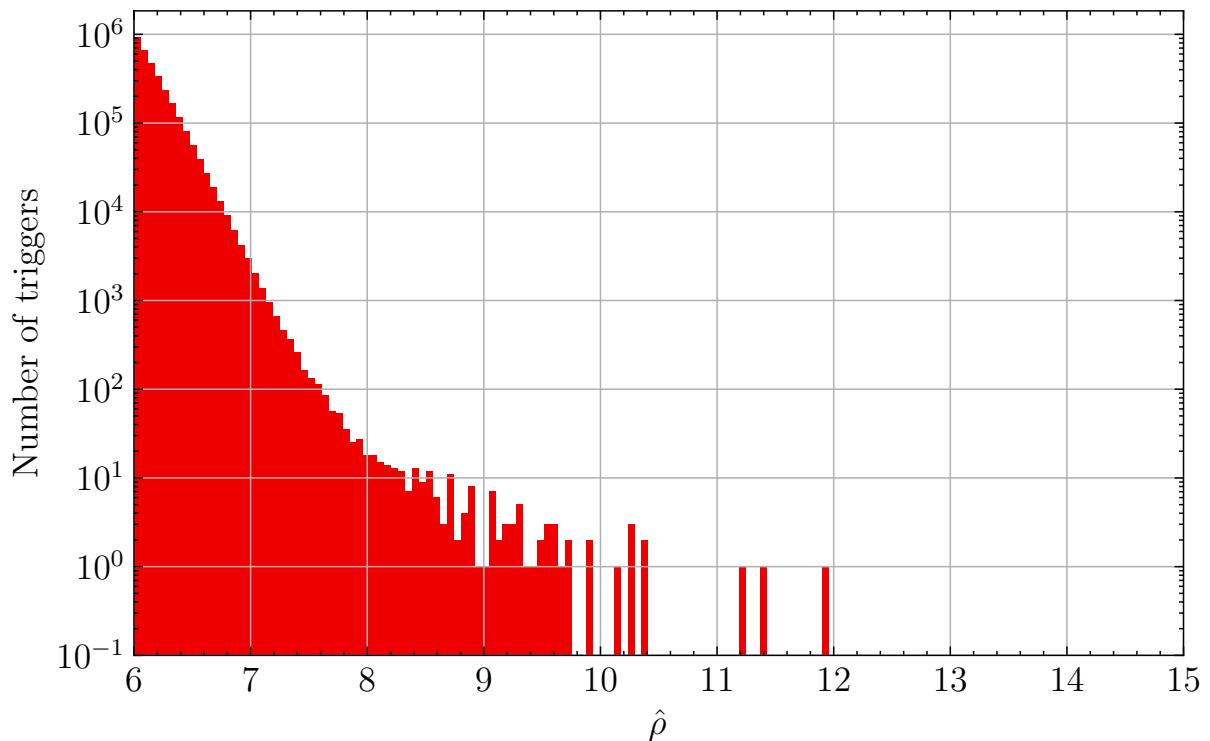


Figure 5.1: The new-SNR,  $\hat{\rho}$ , distribution of LIGO-Hanford triggers for a chunk of data searched through using PyCBC Offline in the third observing run. The higher SNR triggers can be attributed to non-Gaussian noise transients.  $\hat{\rho}$  has been calculated using the Allen- $\chi^2$  [?] and sine-Gaussian  $\chi^2$  [?] to re-weight SNR.



$\rho$  threshold for each template are collected and saved. These triggers are then used to calculate  $\alpha$  and  $\mu$  for each template and these parameters are saved to data files which contain all the template fits for the entire template bank for that chunk of data. To understand how the template fits contributes to the ranking statistic, we must first construct the ranking statistic which includes the exponential noise model. We previously defined a general detection statistic in Equation ?? which was constructed to obtain the quadrature sum statistic for a detector with stationary, Gaussian noise. This ranking statistic models the noise falloff in the distribution as a Gaussian. The new exponential fit to the noise falloff doesn't make this assumption so we return to,

$$\Lambda(\vec{\theta}) = \mu_s \frac{p^S(\vec{\theta})}{p^N(\vec{\theta})}, \quad (5.11)$$

where  $\Lambda$  is the likelihood ratio (Equation ??),  $\mu_s$  is the astrophysical signal rate and is assumed to be constant,  $p^S(\vec{\theta})$  is the signal event rate density (signal rate) and  $p^N(\vec{\theta})$  is the noise event rate density (noise rate). Due to expected values of signal and noise rate spanning many orders of magnitude we take the logarithm of Equation ??,

$$R = \log \Lambda = \log p^S(\vec{\theta}) - \log p^N(\vec{\theta}). \quad (5.12)$$

The noise rate is a combination of the single detector noise rates,

$$p^N(\vec{\theta}) = A_N \prod_d r_d(\hat{\rho}_d), \quad (5.13)$$

where  $A_N$  is the ‘allowed area’ for coincident noise events from two detectors,  $A_{N\{12\}} = 2\tau_{12}$ , where  $\tau_{12}$  is the gravitational-wave travel time window between detectors 1 & 2 with a small allowance for timing error (currently 0.03 seconds) and  $r_d$  is the single detector noise event rate density for each detector  $d$ . The natural logarithm of this noise rate is taken to get the final combined log noise rate,

$$\log(p^N(\vec{\theta})) = \log(A_N) + \sum_d \log(r_d(\hat{\rho}_d)). \quad (5.14)$$

The single detector noise rate is calculated with the product of the count of triggers above the  $\hat{\rho} > 6.0$ ,  $\mu$ , and the probability of  $\hat{\rho}$  given the template (including  $\alpha$  and  $\mu$ ) and noise,

$$r_d(\hat{\rho}_d; \vec{\theta}; N) = \mu(\vec{\theta}) p(\hat{\rho}|\vec{\theta}, N), \quad (5.15)$$

where

$$p(\hat{\rho}|\vec{\theta}, N) = \alpha(\vec{\theta}) \exp \left( -\alpha(\vec{\theta}) \cdot (\hat{\rho} - \hat{\rho}_{thresh}) \right) \quad (5.16)$$



thereby giving the single detector noise rate as,

$$r_d(\hat{\rho}; \vec{\theta}, N) = \mu(\vec{\theta})\alpha(\vec{\theta}) \exp\left(-\alpha(\vec{\theta}) \cdot (\hat{\rho} - \hat{\rho}_{thresh})\right) \quad (5.17)$$

and when the natural logarithm is taken we arrive at the complete single detector log noise rate,

$$\log r_d(\hat{\rho}) = \log \mu(\vec{\theta}) + \log \alpha(\vec{\theta}) - \alpha(\vec{\theta}) \cdot (\hat{\rho} - \hat{\rho}_{thresh}) \quad (5.18)$$

for each detector,  $d$ .

## 5.5 Measuring improvements in search sensitivity

To quantifiably demonstrate that the new additions to the ranking statistic have improved the live search we can investigate the differences in sensitivity when the PSD variation and exponential noise model are included. We quantify the sensitivity of the search by calculating the sensitive volume in which we can observe gravitational-wave signals. The sensitive volume is calculated by measuring the detection efficiency of different distance bins that are taken from an injection set and then multiplying by the volume enclosed by the distance bins. The volume of each bin is then summed to find the total sensitive volume of the search [? ].

We can search through a period of data which contains an injection set with two instances of the PyCBC Live search, one containing the new ranking statistic additions and one representing the original ranking statistic. The ratio of the new statistic sensitive volume to the original statistic sensitive volume will indicate the change in sensitivity when including the new ranking statistic components.

We have chosen to test the ranking statistic changes to PyCBC Live using a smaller template bank than the current PyCBC Live template bank, which contains over 730,000 templates [? ]. PyCBC-BBH is a search focussed on binary black hole (BBH) signals performed in the third observing run [? ? ] and used a template bank with only 15,436 templates with the drawback that we are only considering BBH signals. This smaller template bank uses fewer computational resources and allows a full week of data to be searched through with PyCBC Live in as little as a day. The injection set used in the third observing run is shown in Table ?? [? ] with rows for the specific cuts on injections for both the PyCBC-broad (full parameter space search) and PyCBC-BBH search.

As previously mentioned, the template fits are measured from the triggers found in the previous week to the week currently being searched through. Therefore, to perform our injection set test we need to do an initial search with PyCBC Live us-



		Mass distribution	Mass range ( $M_{\odot}$ )	Spin range
Injections	BBH	$p(m_1) \propto m_1^{-2.35}$ $p(m_2 m_1) \propto m_2$	$2 < m_1 < 100$ $2 < m_2 < 100$	$ \chi_{1,2}  < 0.998$
	NSBH	$p(m_1) \propto m_1^{-2.35}$ Uniform	$2.5 < m_1 < 60$ $1 < m_2 < 2.5$	$ \chi_1  < 0.998$ $ \chi_2  < 0.4$
	BNS	Uniform	$1 < m_1 < 2.5$ $1 < m_2 < 2.5$	$ \chi_{1,2}  < 0.4$
PyCBC-broad	BBH			$\mathcal{M} > 4.353$
	NSBH		$2.176 < \mathcal{M} < 4.353$	
	BNS			$\mathcal{M} < 2.176$
PyCBC-BBH	BBH			$\mathcal{M} > 4.353$

Table 5.1: Recreated from [? ]. The intrinsic parameter distributions used to generate the injection set used for computing the sensitive volume change when including the PSD variation statistic and exponential noise model. Here PyCBC-broad is the PyCBC offline search across the full template bank and parameter space, including templates representing binary black hole, neutron star black hole and binary neutron star signals. This is comparable to the template bank used by the PyCBC Live full-bandwidth search. PyCBC-BBH considers only templates where  $\mathcal{M} > 4.353 M_{\odot}$ , which includes only binary black hole signals. This is the same template bank that is used for the analysis in this chapter.

ing the original ranking statistic to generate a week's worth of triggers and create template fit statistics. The second week of data is then created containing a large number of injected gravitational-wave signals that is searched through with PyCBC Live using both the original ranking statistic and the new ranking statistic. The two weeks of O3b gravitational-wave data used lasted from 06 January 2020 23:59:42 to 20 January 2020 23:59:42 and, according to the gravitational-wave catalogue for the third observing run [? ], there are no known gravitational-wave signals present. After both searches have completed analysing the second week of data we can estimate the sensitivity improvement. We can choose a threshold false-alarm rate (FAR) and count the number of injections in both searches that have been found with a FAR lower than that number, if the counted number of injections is higher when including the new ranking statistic additions we have seen an increase in the sensitivity of the search. We can do this for a range of FAR values from  $10^4 - 10^{-4}$  to build up a picture of the sensitivity ratio curve.



## 5.6 Improvements in sensitivity

By adding the PSD variation and the exponential noise model to the ranking statistic we expect to find gravitational-wave injections with a greater significance than before. This significance is determined by the FAR of the injection, which has a direct mapping to the ranking statistic value, an event found with a FAR of 1 per year would indicate that an event with the exact same parameters is expected to be seen in the data caused by non-astrophysical noise-not a real event-once per year of analysis time. We commonly use the inverse false-alarm rate (IFAR) with units of *years* where a large IFAR indicates a more confident detection. An IFAR threshold is chosen to separate events which are real from those that are false alarms. A balance must be made to avoid contaminating the detection catalogue with a large number of potential false alarms, while not missing any real events. In this analysis we are considering only two detectors, allowing potentially more false-alarms from coincident noise when compared to a three detector configuration and therefore we have chosen a more conservative IFAR threshold of 1 year to assess sensitivity improvements. The current threshold for LIGO/Virgo alert generation is 1 per 2 months [?] so low-latency gravitational-wave searches will identify more gravitational-wave events, but these detections have additional post detection checks from data quality checks from detector characterisation [?] and parameter estimation [?] which can also rule out false-alarms and to prevent catalogue contamination.

To measure the sensitivity improvement of the new ranking statistic we count the number of injections found by both searches over a range of IFAR values from  $10^{-4} - 10^4$  and take the ratio of the counts. The injection set is pre-weighted to ensure each injection represents an equal volume, therefore instead of distance binning, as described in Section ??, we can simply count the number of injections seen by each search. Figure ?? shows the sensitivity ratio plot comparing the new statistic search to the original statistic search. We see 35% more injections with an IFAR greater than our threshold of 1 year which is a very large increase in sensitivity, however at a FAR of 2 per month we see a sensitivity decrease of 10%. To understand the shape of the sensitivity curve we can look at the new additions to the ranking statistic and understand further how they are changing the significance of individual injections.

## 5.7 Changes in injection significance

The ranking statistic assigned to each event can be mapped directly to FAR, shown in Figure ???. Changing the ranking statistic calculation will lead to incomparable

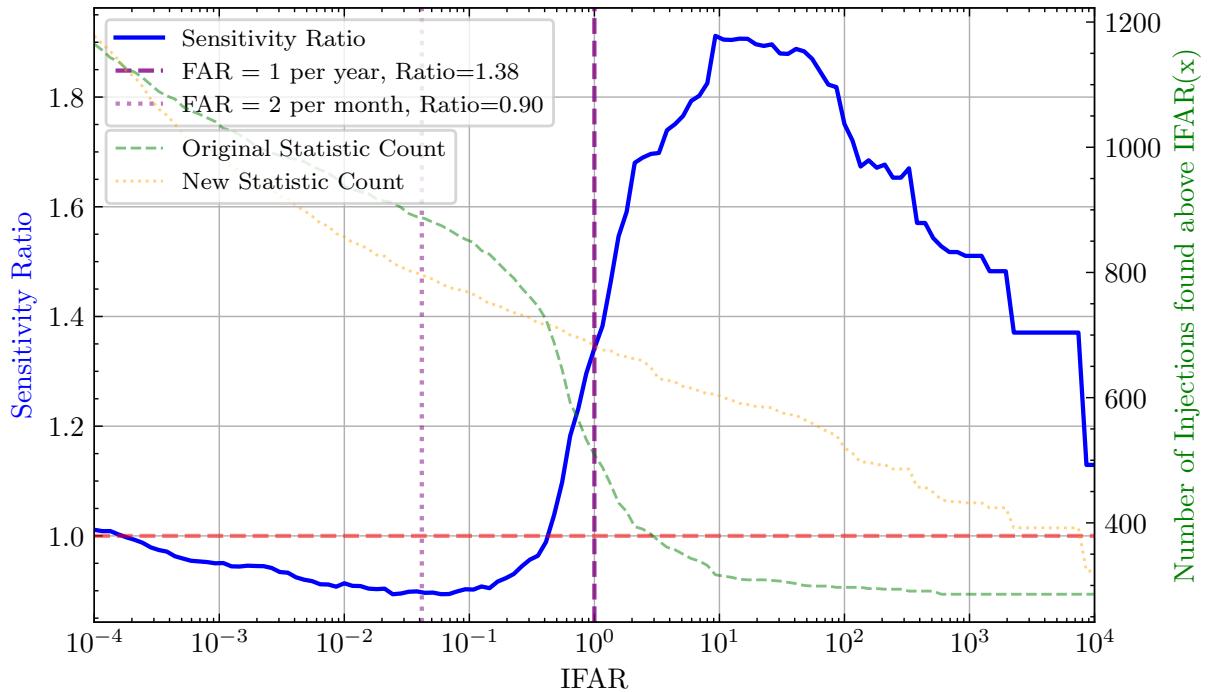


Figure 5.2: The sensitive volume ratio between a PyCBC Live search whose ranking statistic contains PSD variation and an exponential noise model and a PyCBC Live search using the original ranking statistic used by PyCBC Live during the third observing run as a function of inverse false-alarm rate (IFAR). The blue curve represents the sensitivity ratio computed by comparing the number of injections detected with an IFAR exceeding the corresponding x-axis value. An increase in sensitivity is indicated by values greater than 1, the horizontal red-dashed line. Two vertical dashed lines at fixed false-alarm rates (FAR) highlight specific thresholds: FAR = 1 per year (dark purple, dashed) with a sensitivity ratio of 1.38, and FAR = 2 per month (light purple, dotted) with a sensitivity ratio of 0.90. The green and orange curves represent the cumulative number of injections detected in the old and new searches, respectively, as a function of IFAR.

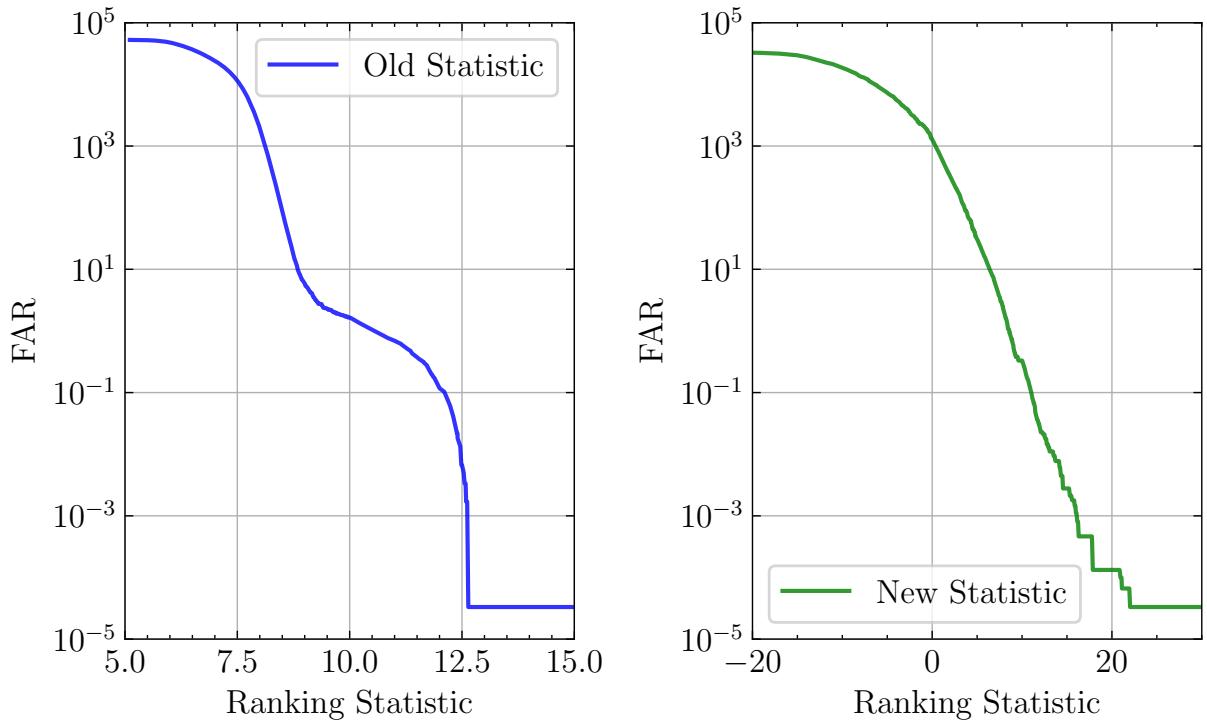


Figure 5.3: The mapping between false-alarm rate (FAR) and ranking statistic for the original PyCBC Live ranking statistic used in the third observing run (left, blue) and the new PyCBC Live ranking statistic including PSD variation and an exponential noise model (right, green). Ranking statistic values cannot be compared between ranking statistics, but injection FARs can be. Higher ranking statistic values indicate better, lower FARs. The original ranking statistic has a narrower range of allowed ranking statistic values (5 – 12.5) along with a distinct curvature whereas the new statistic ranking statistic can take a broader range of ranking statistic values (–20 – 22) and has a relatively smooth drop off with increasing ranking statistic values. The ‘shoulder’ in the original ranking statistic was an unwanted feature which has now been removed.



values between searches, but the individual injection IFAR values can be compared. The PyCBC Live search will matched filter the template bank and the data and where the resulting  $\rho$  and  $\hat{\rho}$  is above pre-defined thresholds (both 4.5 in the third observing run) the trigger is kept. The coincidence ranking statistic value is calculated and then a ranking statistic value is assigned to the event. The initial  $\rho$  values are the same between the two searches because these depend only on template and data but  $\hat{\rho}$  depends on the single detector ranking statistic which has the addition of PSD variation in our new statistic, therefore it is possible for the PSD variation to prevent the saving of a trigger due to a trigger's  $\hat{\rho}$  being re-weighted below the threshold.

We can plot the IFAR value of each injection that has been found in both searches, seen in Figure ???. Within Figure ?? we have plotted a line on the diagonal  $y = x$ , injections above this line ( $y > x$ ) have been found with a larger IFAR with the new ranking statistic and injections below ( $y < x$ ) have been found with a lower IFAR with the new statistic. Figure ?? highlighted the decrease in sensitivity at a FAR of 2 per month, this decrease is due to an excess number of injections in the red-shaded box in the bottom-right of Figure ??, this region represents injections which have been found by the original ranking statistic below a FAR of 2 per month but with a FAR above 2 per month when found by the new ranking statistic. We now proceed to explore these results, investigating why the injections found in the red-shaded box have been down-ranked and other interesting features.

### 5.7.1 Effects of PSD variation on sensitivity

The injections shown in Figure ?? that lie below the  $y = x$  diagonal line have been found with a lower significance when including the PSD variation and the exponential noise model in the ranking statistic. We can separate the new additions and investigate the individual contributions to the changes in sensitivity.

The PSD variation statistic measures the non-stationary noise in the data and applies a weighting to the trigger  $\rho$  value calculated by the matched filter of template and data, prior to the calculation of  $\hat{\rho}$ . To understand the effect of the PSD variation statistic on the injection significance we can compare the IFAR of injections found by the new statistic which includes both the PSD variation statistic and the exponential noise model and the IFAR of the same injections but with only the addition of the exponential noise model. This comparison is shown in Figure ?? In this comparison, 550 injections were found with a higher significance when **not** including PSD variation, 399 were found with a higher IFAR when including PSD variation and 326 were found with the same IFAR. We can also observe a greater dispersion of injections

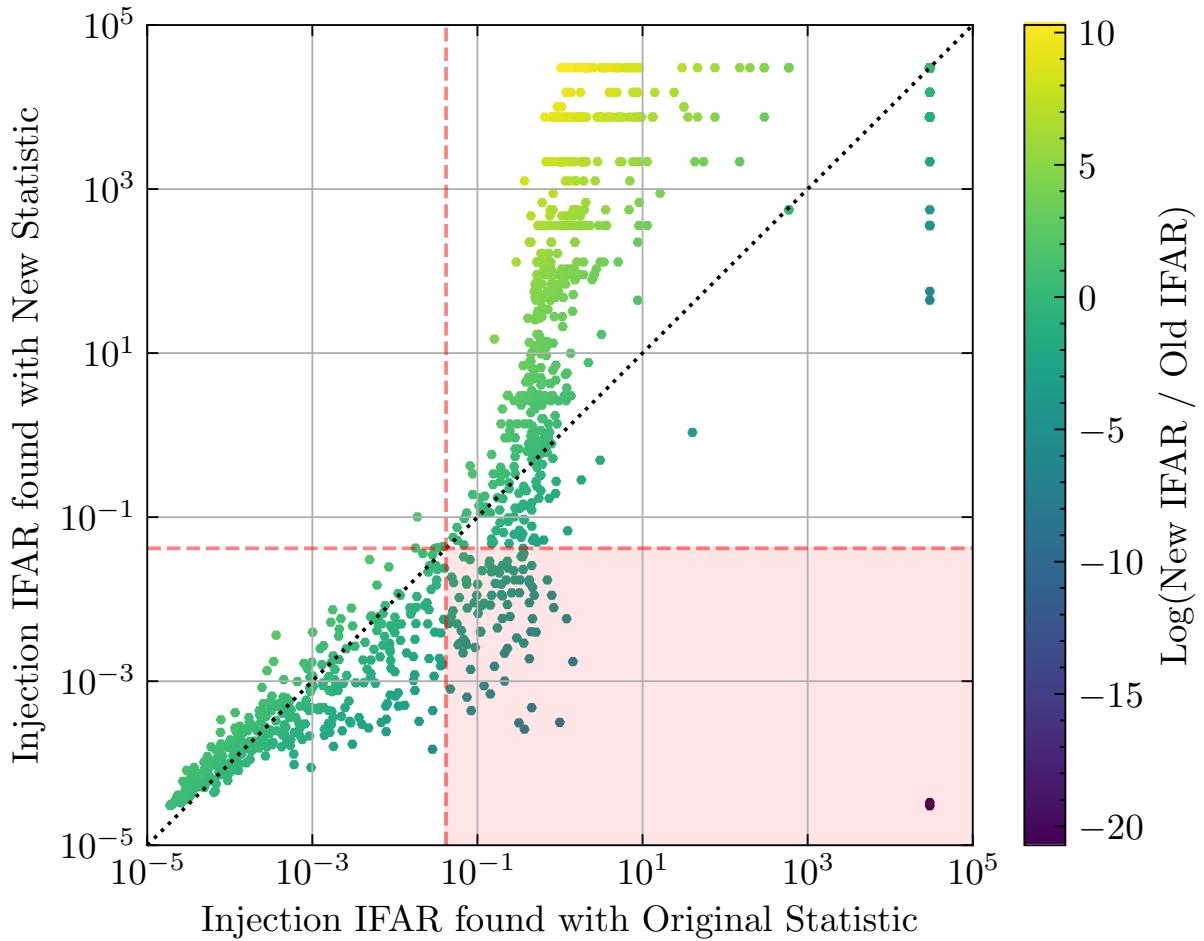


Figure 5.4: The inverse false-alarm rate (IFAR) values recorded for each injection that was seen by both the PyCBC Live search that includes the PSD variation and exponential noise model in the ranking statistic and the PyCBC Live search using the same ranking statistic as used in the third observing run. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded using the original ranking statistic and the y-axis showing the IFAR recorded using the new ranking statistic. The colour bar indicates the logarithmic difference in IFAR between the two searches, with positive values (yellow) indicating better performance of the new statistic and negative values (purple) indicating a worse performance with the new statistic. The dashed line at  $y=x$  represents equal IFAR values for both searches. The vertical and horizontal red dashed lines denote a false-alarm rate threshold of 2 per month. The highlighted region in the bottom right of the plot indicates excess injections that have been seen by the original ranking statistic above the 2 per month threshold and by the new ranking statistic below the 2 per month threshold.

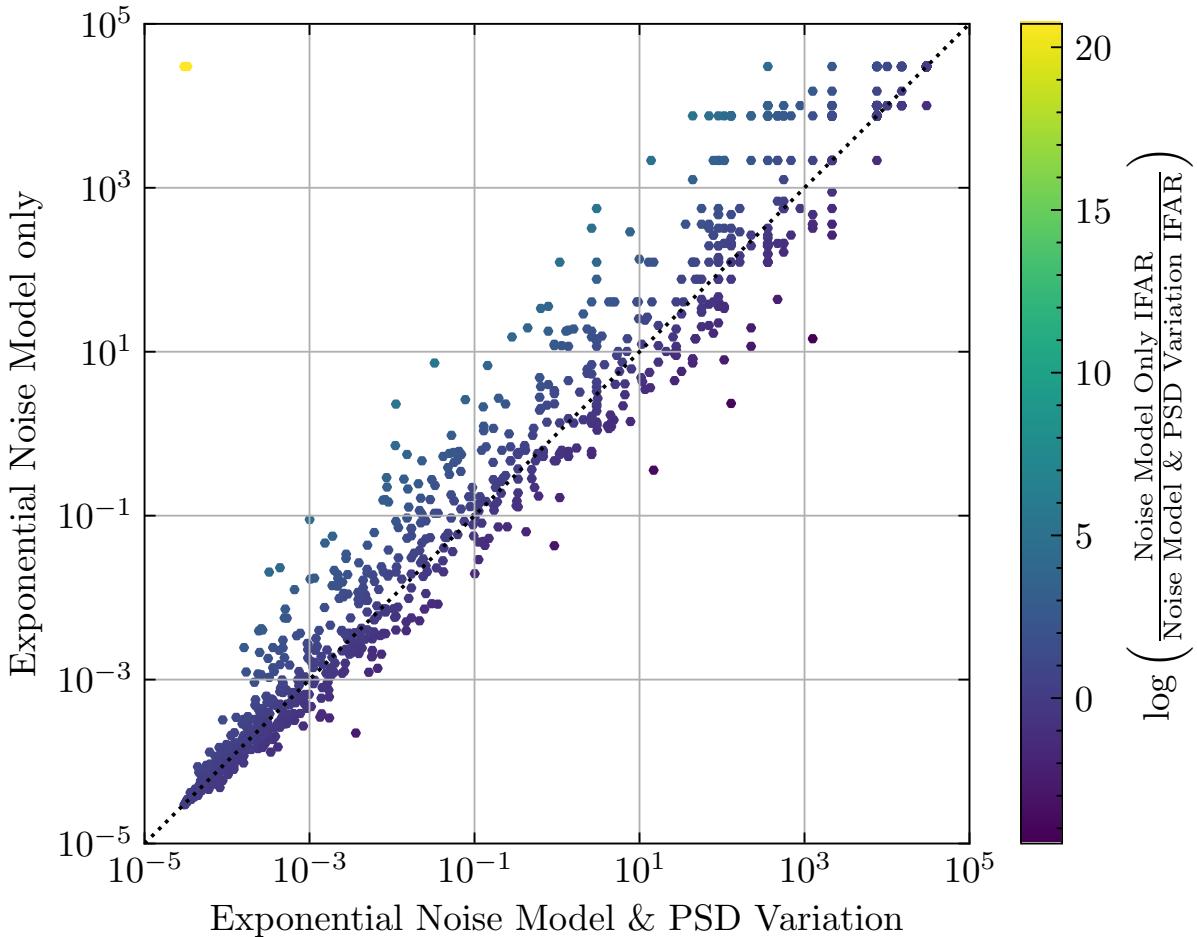


Figure 5.5: The inverse false-alarm rate (IFAR) values recorded for injections that were seen by both the PyCBC Live search that includes new additions of the PSD variation statistic and exponential noise model in the ranking statistic and the PyCBC Live search which has new additions of only the exponential noise model. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded by the PyCBC Live search including the exponential noise model and PSD variation ranking statistic and the y-axis showing the IFAR recorded by the PyCBC Live search which has added only the exponential noise model. The colour bar indicates the logarithmic difference in IFAR between the two searches, with positive values (yellow) indicating better performance when **not** including PSD variation and negative values (purple) indicating a better performance with the PSD variation statistic. The dashed line at  $y=x$  represents equal IFAR values for both searches.

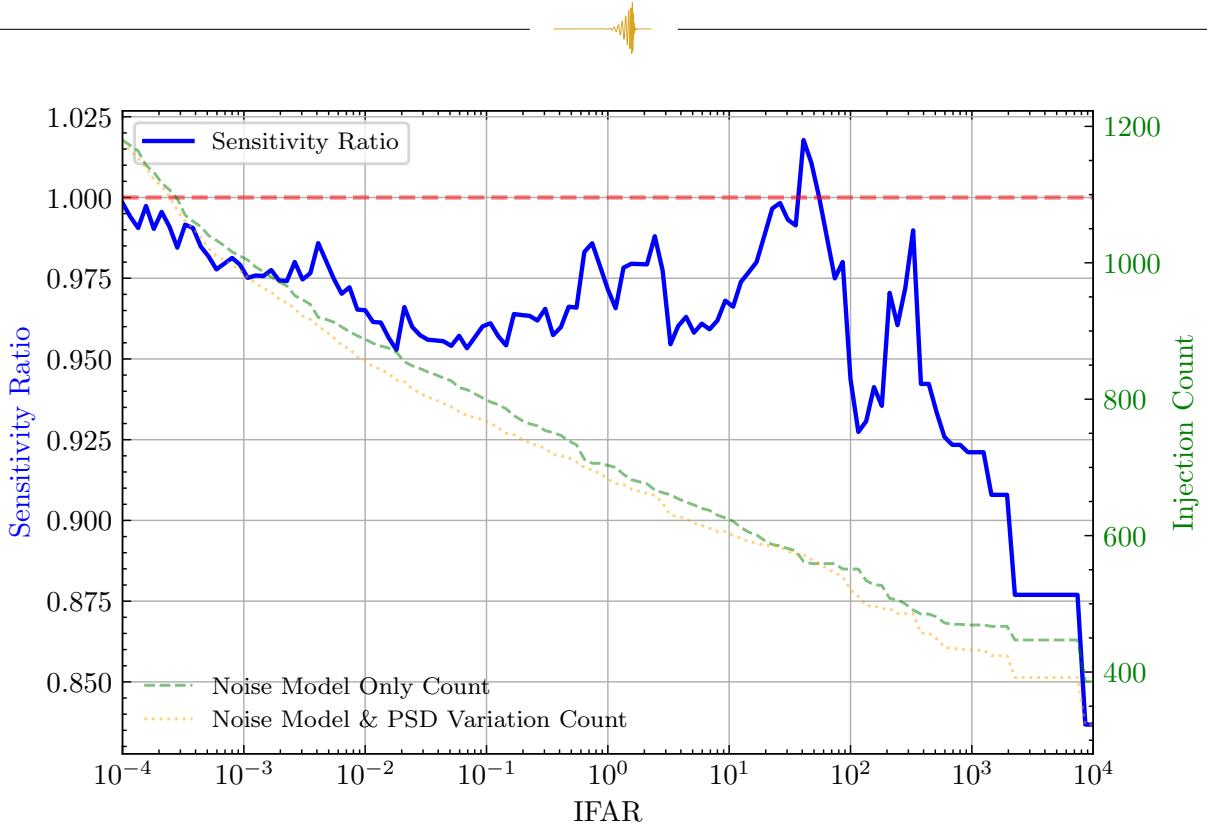


Figure 5.6: A comparison of sensitive volume ratio between two PyCBC Live searches: one which includes the addition of the PSD variation statistic and an exponential noise model in the ranking statistic and one which includes only the addition of the exponential noise model. The blue curve represents the sensitivity ratio computed by comparing the number of injections detected with an inverse false-alarm rate (IFAR) exceeding the corresponding x-axis value. An increase in sensitivity is indicated by values greater than 1, the horizontal red-dashed line. The green dashed curve represents the cumulative number of injections detected as a function of IFAR by the PyCBC Live search with added PSD variation and the exponential noise model. The orange dashed curve represents the cumulative number of injections detected as a function of IFAR by the PyCBC Live search with only the exponential noise model added.

found above the diagonal line when compared to those found below it, meaning injections found above the diagonal line are found with a greater significance difference than those found below the line. Alongside this general trend we can see a few injections in the top-left of the plot which have had a very large increase in significance when we do not include PSD variation in the statistic, these injections correspond to the heavily down-ranked injections in the bottom-right of Figure ??.

We can compare the sensitivity change when including the PSD variation statistic by plotting the sensitivity curve in Figure ??-. From this, it is clear to see there is a small decrease in sensitivity when including the PSD variation statistic in the new ranking statistic. We note that the current implementation of PSD variation in PyCBC



Offline has problems, specifically, PSD variation is eliminating extremely significant low-mass injections. Investigation into the PSD variation issues is being conducted and future iterations of PSD variation may improve its performance in PyCBC Live.

We have previously described how the PyCBC Live search re-calculates the PSD every analysis segment and replaces the currently used PSD when the BNS distance of the new PSD differs from the current PSD's by more than 1%. This reduces the need to include the PSD variation statistic to tackle non-stationary noise in the data. In light of the sensitivity decrease when including PSD variation, we make the choice to not use PSD variation in the ranking statistic going forward.

When adding only the exponential noise model to the ranking statistic we can re-analyse the sensitivity improvements to produce Figure ???. This shows a greater increase across the IFAR range, with sensitivity increase at a FAR of 1 per year of 43% and a decrease of 6% at a FAR of 2 per month.

We can plot the IFAR vs IFAR plot for the search including only the exponential noise model in the ranking statistic compared to the original search, seen in Figure ??, and when we compared to Figure ?? we can see that the distribution of injections at low IFARs has decreased in dispersion around the diagonal line. We have highlighted three regions of interest which contains injections that have decreased in significance that we will investigate further and understand the cause of their down-ranking.

## 5.7.2 Templates which trigger frequently with large SNRs

The new exponential noise model in the ranking statistic allows us to model individual template noise distributions using that template's historical triggers. This allows us to down-rank templates which frequently trigger on non-Gaussian noise with a high  $\hat{\rho}$ . As describe in Equation ??, the exponential noise model is characterised by an exponential fit to the slope of the trigger distribution and is quantified by the exponential fit factor,  $\alpha$ , and the count of triggers above  $\hat{\rho}_{thresh}$ ,  $\mu$ . Therefore, using these two parameters we must be able to differentiate between templates which trigger frequently on noise with large  $\hat{\rho}$  and those that rarely trigger on noise. All templates in the bank will trigger on both Gaussian noise and glitches in our data, and after applying all signal consistency tests these triggers might still might have  $\hat{\rho} > 8.0$ .

Figure ?? shows an example of the exponential fitting to a group of template's noise distributions to obtain the exponential fit factors. This group of templates have effective spin,  $\chi_{eff} = -0.333$  to  $0.333$ , symmetric mass ratio  $\eta = 0.208 - 0.229$  and the different template duration bins and corresponding exponential fitting factors have been plotted over the top. This figure shows the difference in exponential fit

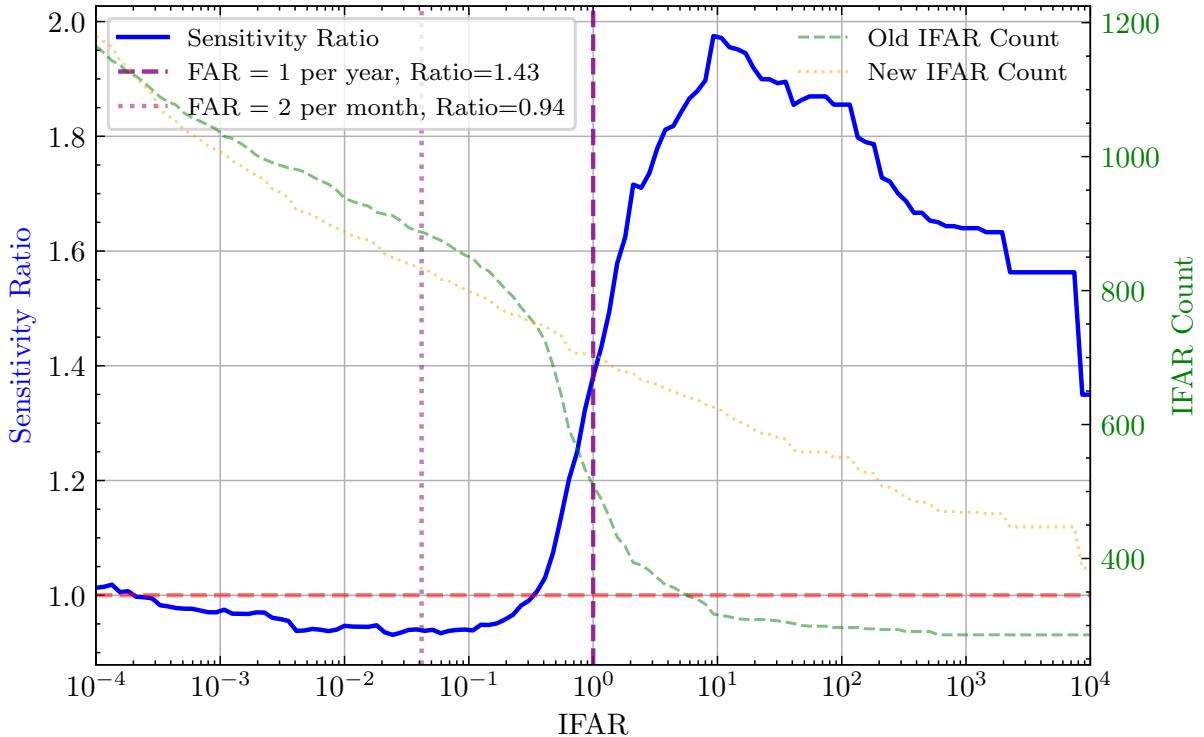


Figure 5.7: A comparison of sensitive volume ratio between a PyCBC Live search whose ranking statistic contains an exponential noise model and a PyCBC Live search using the original ranking statistic used by PyCBC Live during the third observing run as a function of inverse false-alarm rate (IFAR). The blue curve represents the sensitivity ratio computed by comparing the number of injections detected with an IFAR exceeding the corresponding x-axis value. An increase in sensitivity is indicated by values greater than 1, the horizontal red-dashed line. Two vertical dashed lines at fixed false-alarm rates (FAR) highlight specific thresholds: FAR = 1 per year (dark purple, dashed) with a sensitivity ratio of 1.43, and FAR = 2 per month (light purple, dotted) with a sensitivity ratio of 0.94. The green and orange curves represent the cumulative number of injections detected in the old and new searches, respectively, as a function of IFAR.

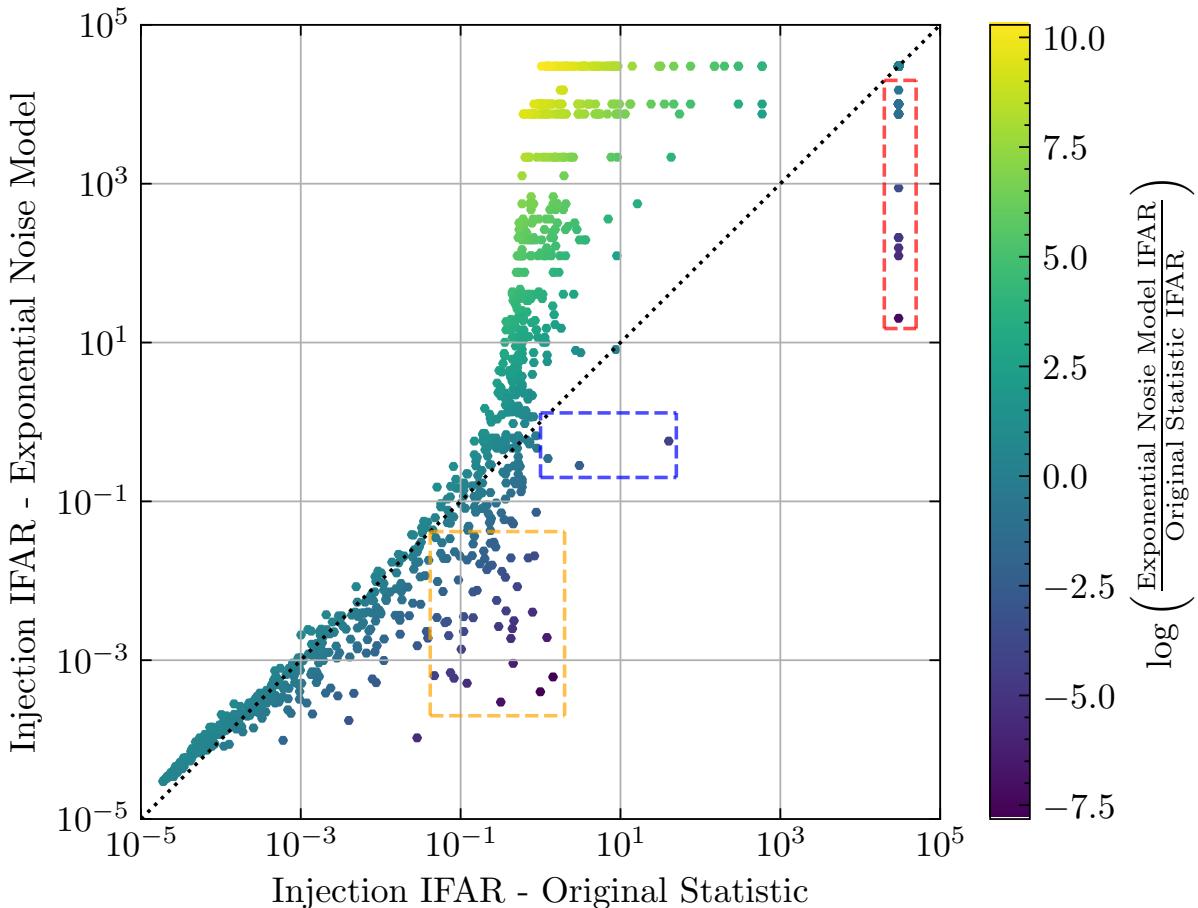


Figure 5.8: The inverse false-alarm rate (IFAR) values recorded for each injection seen by both the PyCBC Live search that includes the exponential noise model in the ranking statistic and the PyCBC Live search using the same ranking statistic used in the third observing run. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded using the original ranking statistic and the y-axis showing the IFAR recorded using the exponential noise model ranking statistic. The colour bar indicates the logarithmic difference in IFAR between the two searches, with positive values (yellow) indicating better performance when including the exponential noise model in the statistic and negative values (purple) indicating a better performance with the original statistic. The dashed line at  $y=x$  represents equal IFAR values for both searches. Three regions containing injections that have been down-ranked and will be investigated have been highlighted in dashed boxes.

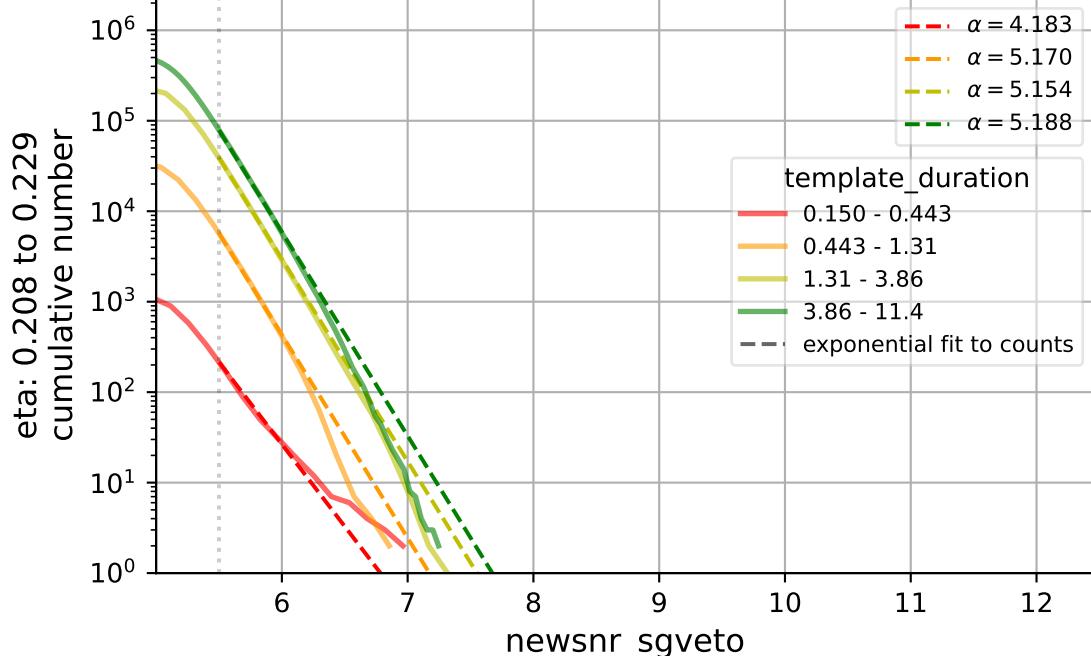


Figure 5.9: The noise distribution and template fits for a bin which contains triggers from templates which have symmetric mass ratio,  $\eta$ , between  $0.208 - 0.229$  and effective spin,  $\chi_{eff}$ , between  $-0.333 - 0.333$ . The template triggers are further split into 4 bins depending on the duration of the template responsible for the trigger (in seconds). The solid coloured lines represent the cumulative counts of triggers that have been found in a template duration bin with  $\rho$  that has been re-weighted by the traditional  $\chi^2$  [?] and the sine-Gaussian  $\chi^2$  [?] to obtain new SNR,  $\hat{\rho}$ . The dashed coloured lines represent the exponential fit that has been made to the cumulative counts for each template duration bin where the exponential fit factor value,  $\alpha$ , is displayed in the top right most legend.



slopes for the varying template bins and it can be seen that the bins which trigger more frequently with a higher new SNR have shallower slopes (lower  $\alpha$ ) compared to the steeper slopes (higher  $\alpha$ ) in other bins. Equation ?? shows that higher  $\alpha$  values correspond to lower noise rates and lower  $\alpha$  and higher  $\mu$  values produce higher noise rates.

Templates which trigger more frequently and with a large  $\hat{\rho}$  on noise will be down ranked by the ranking statistic, but real gravitational-wave signals that match this template will also be down ranked if they have low single detector  $\hat{\rho}$ . We can plot Equation ??, which calculates the single detector noise rate, in Figure ?? to show the dependence on  $\alpha$  and  $\mu$ . In our search  $\alpha$  can take values 1.5 – 6.0 and  $\mu$  from 5 – 45 and we have used a static  $\hat{\rho}$  of 10.0 in the equation. It can be seen that a higher  $\alpha$  value has a greater effect on the noise rate and the contribution from  $\mu$  has a maximum value of  $3.8, + \log \mu$  in Equation ???. Figure ?? shows the noise rate's dependence on  $\alpha$  and  $\hat{\rho}$  while  $\mu$  has been set to the benchmark noise rate of  $-14.6$ , which is the representative noise rate measured empirically during the second observing run. A black dashed line has been plotted at  $\hat{\rho} = 6.0$  due to triggers below this  $\hat{\rho}$  threshold all being assigned  $\alpha = 6.0$ , therefore the noise rate below this threshold is constant for all  $\alpha$  values on the plot. We can clearly see a strong correlation between  $\hat{\rho}$  and noise rate with a heavy dependency on  $\alpha$ , to the point where high  $\hat{\rho}$  triggers with low  $\alpha$ —while still significant—have higher noise rates than triggers with high  $\alpha$ .

We can plot the IFAR vs IFAR distribution for the injection set for a single detector's  $\alpha$ ,  $\mu$  and log noise rate to see if there is any clear dependence on the template fit parameters, seen in figures ???. It is not clear from these figures that  $\alpha$  (top-left) and  $\mu$  (top-right) are having a large effect on the log noise shown in the bottom-left figure. There is a curve of high  $\alpha$  points above the  $y = x$  line in the top-left figure but no clear distinction between higher and lower  $\alpha$  values corresponding to higher and lower IFAR values. As mentioned before and reinforced by the top-right plot,  $\mu$  has a small effect on the noise rate and low  $\mu$  values are more common than high ones.

Figure ?? shows the combined detector noise rates (Equation ??), where we can see a clearer distinction between high and low noise rates and how this is affecting the IFAR values. However, looking at this plot we can see that combined noise rate does not completely explain the improved significance for a number of injections, we can see clearly there are some low noise rate injections below the  $y = x$  line which should have caused an increase in the significance, but we have seen a decrease. Therefore, we must investigate other changes to the ranking statistic.

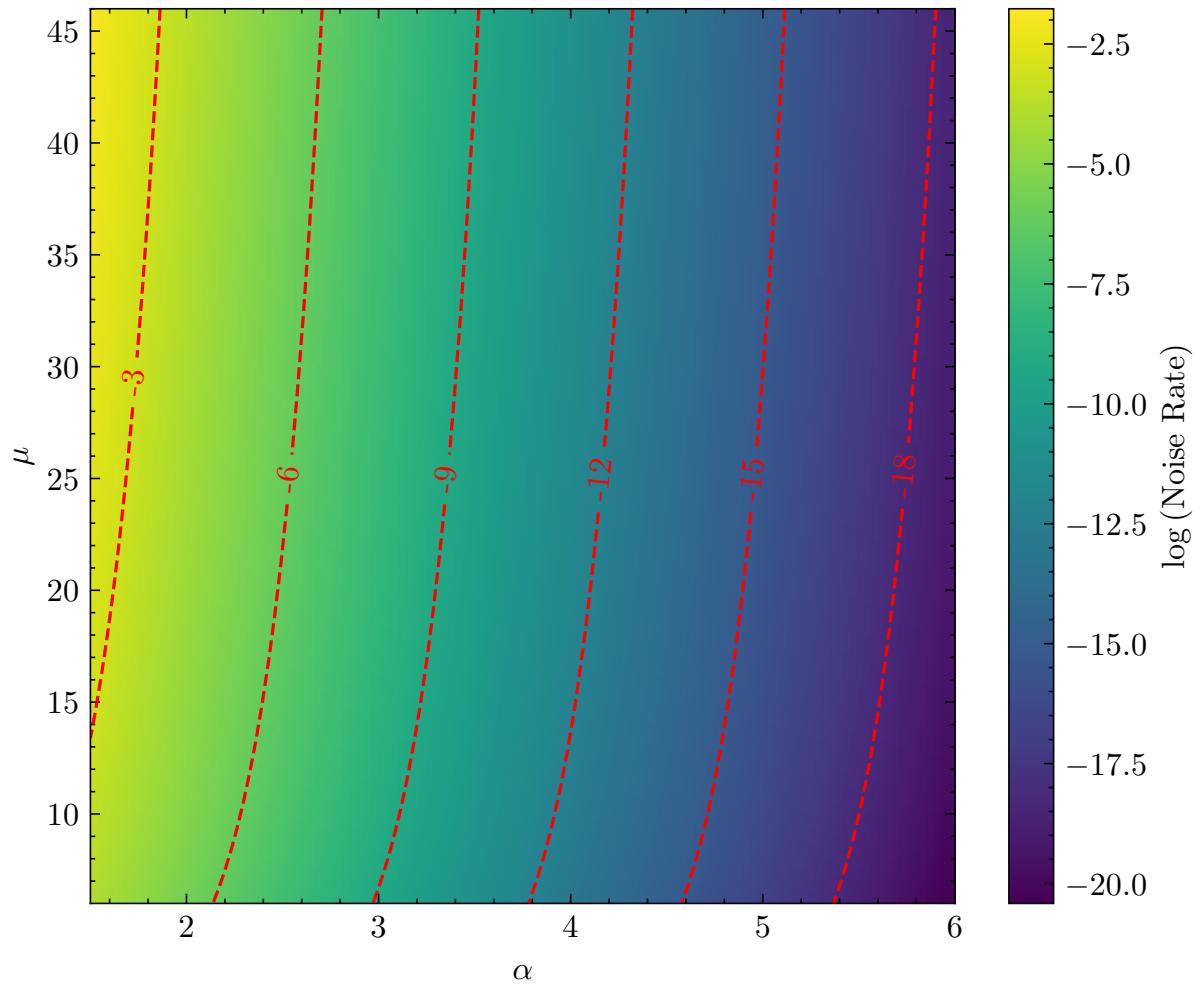


Figure 5.10: The single detector (log) noise rate for varying combinations of exponential fit factor,  $\alpha$ , and trigger rates above a single detector statistic value ( $\hat{\rho}_{thresh} = 6.0$ ),  $\mu$ . Constant contours of single detector noise rate are shown as dashed red lines, with the single detector noise rate value labelled within the contour line. This figure has been made using Equation ?? where  $\hat{\rho} = 10.0$ .

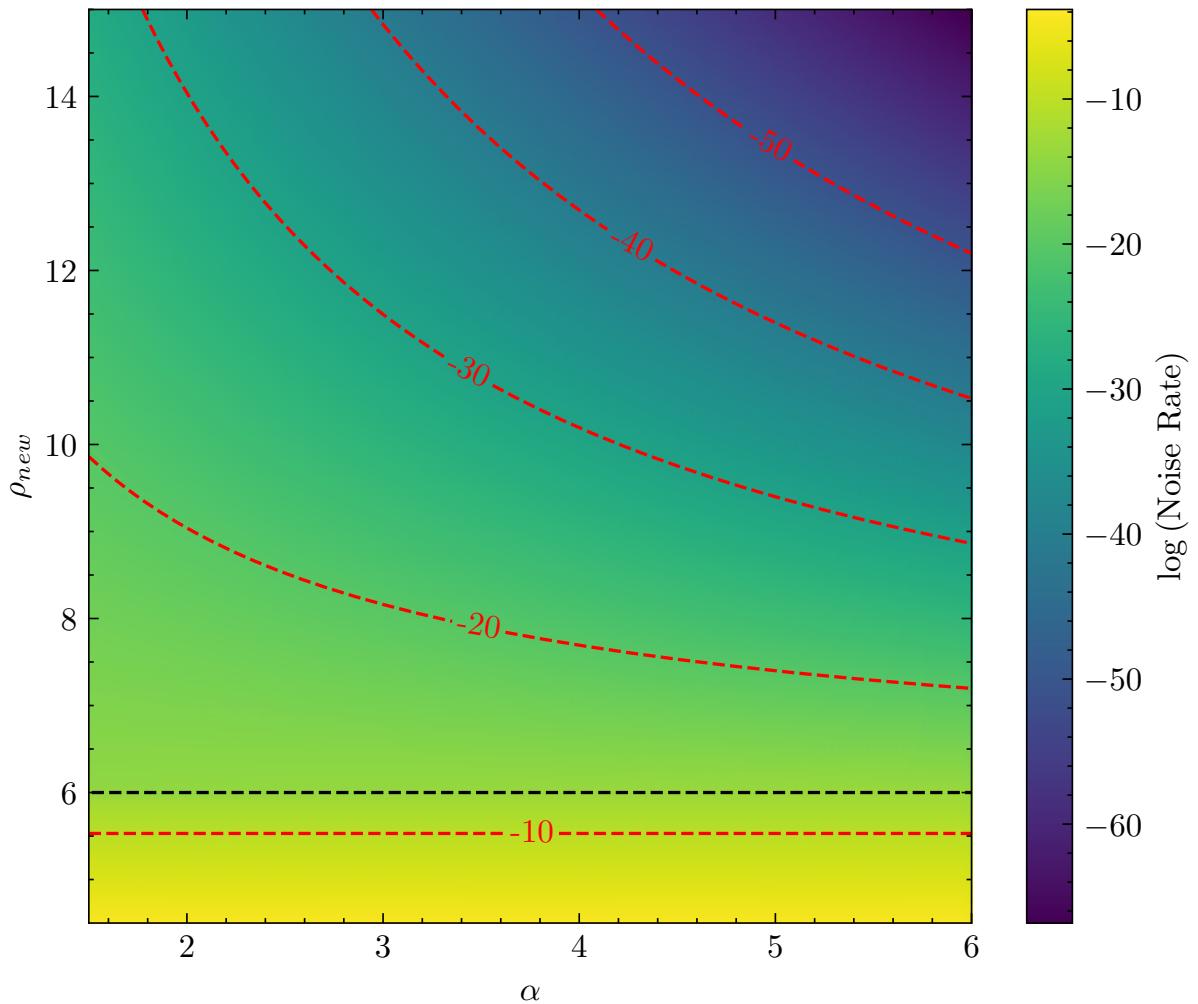


Figure 5.11: The single detector (log) noise rate for varying combinations of exponential fit factor,  $\alpha$ , and single detector statistic value,  $\hat{\rho}$ . Constant contours of single detector noise rate are shown as dashed red lines, with the single detector noise rate value labelled within the contour line. This figure has been made using Equation ?? where  $\mu$  has been set equal to the benchmark noise rate  $-14.6$ .

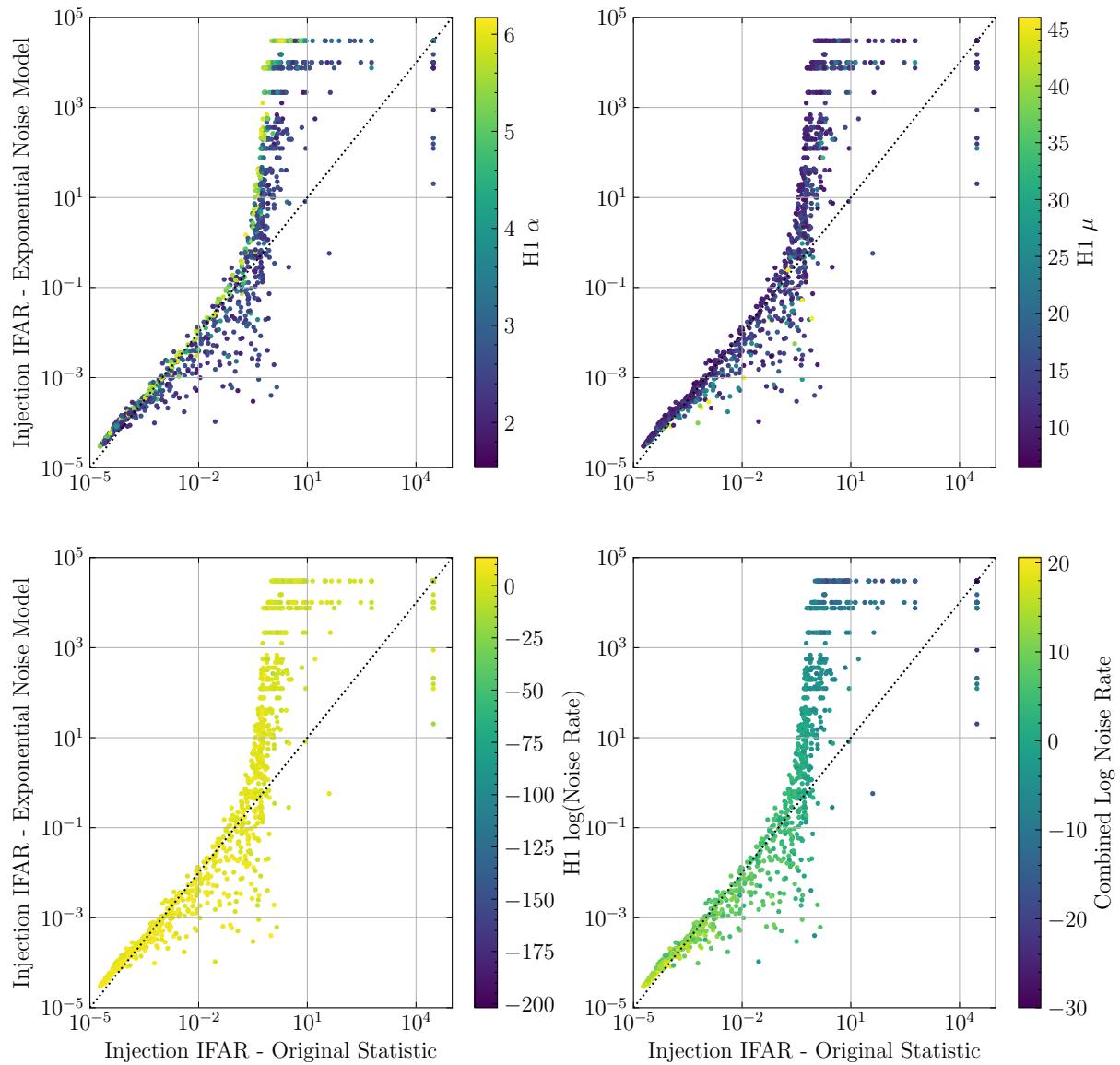


Figure 5.12: The inverse false-alarm rate (IFAR) values recorded for each injection seen by both the PyCBC Live search that includes the exponential noise model in the ranking statistic and the PyCBC Live search using the same ranking statistic used in the third observing run. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded using the original ranking statistic and the y-axis showing the IFAR recorded using the exponential noise model ranking statistic. The top left plot is coloured by the LIGO-Hanford trigger template's exponential fit factor,  $\alpha$ . The top right plot is coloured by the LIGO-Hanford trigger template's count of triggers found above  $\hat{\rho}_{threshold} = 6.0$ ,  $\mu$ . The bottom left plot is coloured by the LIGO-Hanford single detector (log) noise rate, calculated using Equation ???. The bottom right plot is coloured by the combined (log) noise rate, Equation ???.



### 5.7.3 Ranking statistic construction

The original ranking statistic and our new ranking statistic both take contributions from the signal event rate density,  $p^S(\vec{\theta})$ , and the noise event rate density,  $p^N(\vec{\theta})$ . The construction of these ranking statistics has been detailed in sections ?? and ?? where it can be seen they include the contributions from  $p^S(\vec{\theta})$  differently. For comparison, the original statistic is constructed as,

$$R = \sqrt{\hat{\rho}_H^2 + \hat{\rho}_L^2 + 2 \log(p^S(\vec{\theta}))} \quad (5.19)$$

and the statistic including the exponential noise model is,

$$R = \log p^S(\vec{\theta}) - \log(A_N) + \sum_d \log(r_d(\hat{\rho})), \quad (5.20)$$

where  $p^N(\vec{\theta})$  has been substituted with the combined noise rate defined in Equation ??.

The general detection statistic (Equation ??) was constructed so that in stationary, Gaussian noise the ranking statistic recovers the quadrature sum ranking statistic. When including the exponential noise model we are explicitly modelling the detector noise and therefore have chosen to use the optimal detection statistic (Equation ??) which is simply the ratio of signal and noise event rate densities.

Equations ?? and ?? show that the original statistic quadratically adds signal rate and noise rate whereas the new noise distribution model ranking statistic will add signal and noise rates linearly, changing the ratio of contributions from signal rate and noise rate in the ranking statistic calculation.

One particular way the different noise rate models manifests is in the ratio of the two detector  $\hat{\rho}$  values. Modelling the noise distribution falloff as Gaussian causes the ranking statistic to attempt to maximise the squared sum of the  $\hat{\rho}$  values found by each detector with only the signal-consistency tests to re-weight  $\rho$ . Our new model adds additional weights in the form of the template fits so that if a different trigger is seen with slightly lower  $\hat{\rho}$  in both detectors, but the template has a lower noise rate (by having higher  $\alpha$  and lower  $\mu$ ), then it will be preferred by the noise rate model. Figure ?? shows the injections coloured by the SNR ratio in the new statistic search. It can be seen that there are a number of points in regions of interest that have been down-ranked by the new search specifically because they have a high SNR ratio in the old statistic.

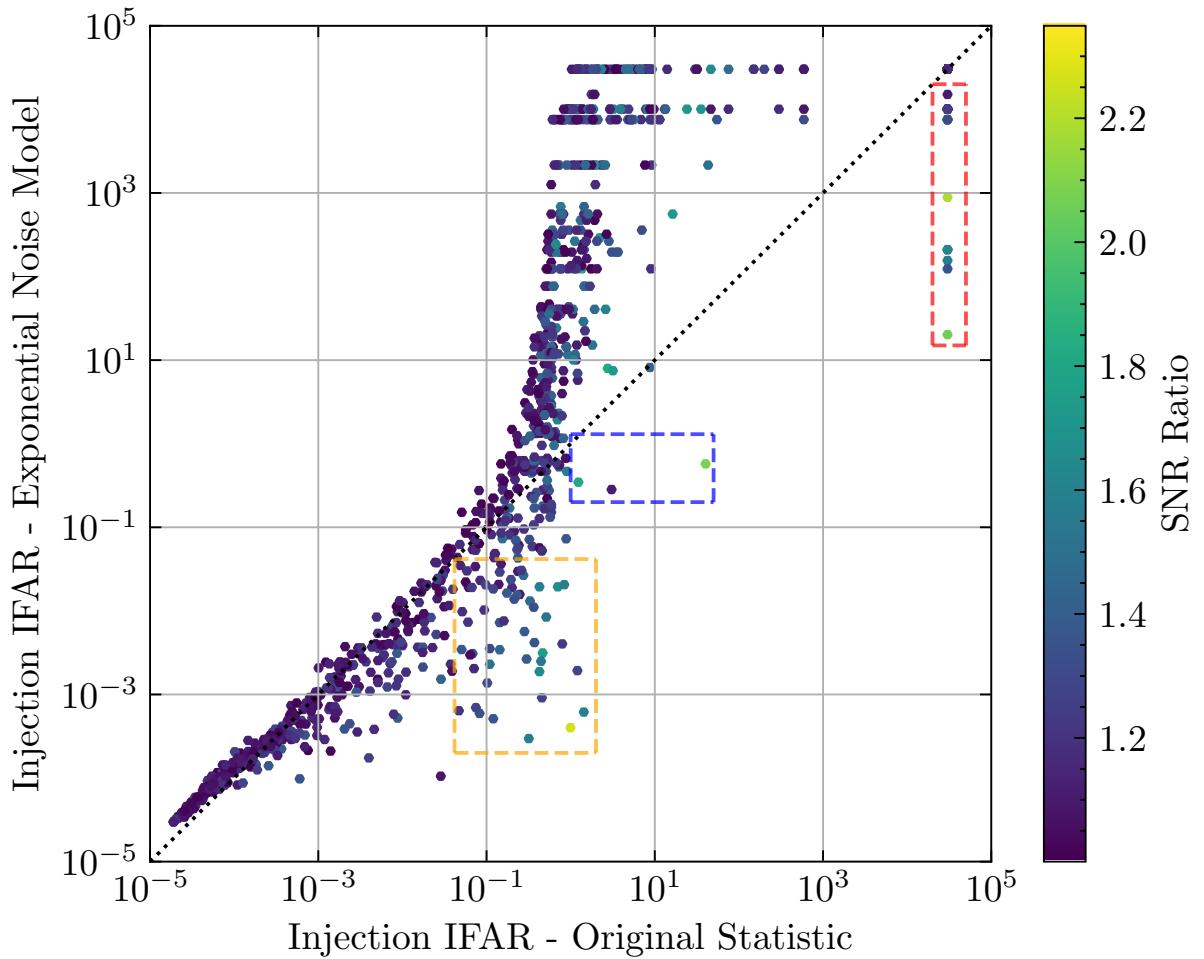


Figure 5.13: The inverse false-alarm rate (IFAR) values recorded for each injection seen by both the PyCBC Live search that includes the exponential noise model in the ranking statistic and the PyCBC Live search using the same ranking statistic used in the third observing run. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded using the original ranking statistic and the y-axis showing the IFAR recorded using the exponential noise model ranking statistic. The injections are coloured by the SNR ratio of the two detector  $\hat{\rho}$  values, set so that the SNR ratio is always greater than 1. The regions highlighted by dashed boxes contain down-ranked injections and are investigated in subsequent sections.



### 5.7.4 Different triggers selected by each search

The ranking statistic is responsible for determining the most significant triggers in an analysis segment in the live search. As stated in Equation ?? the ranking statistic is composed of two primary contributions, the signal rate and the noise rate. As described in the previous section (??), both the signal and the noise rate are template dependent and therefore when the construction of the ranking statistic is changed a different template might become responsible for the preferred trigger compared with the template chosen by the original statistic search. We investigate some injections which have now been found with a different trigger when compared to the original ranking statistic search to understand how changing the ranking statistic individually changes the injections.

Each injection in the initial injection set has an assigned index number. When a trigger is found by the search the identification number of the template responsible is saved with the trigger. Using these two values we can directly compare the template ID of each injection index to find injections which have been found with a different template ID and therefore a different trigger in both searches. Table ?? shows an injection, with injection index = 200, that has been found with a different trigger in both searches. This table displays the values for the signal rate,  $\log p^S(\vec{\theta})$ , the noise

Injection Index = 200	Original Trigger	New Trigger
$\log p^N(\vec{\theta})$	-15.53	-18.79
$\log p^S(\vec{\theta})$	-15.77	-15.67
$R_{new}$	-0.24	3.12

Table 5.2: The (log) signal rate,  $\log p^S(\vec{\theta})$ , (log) noise rate,  $\log p^N(\vec{\theta})$ , ranking statistic value,  $R_{new}$  and inverse false-alarm rate (IFAR), calculated using the ranking statistic which includes an exponential noise model, for the preferred trigger found by the PyCBC Live search when using the original ranking statistic used during the third observing run (Original Trigger) and the different preferred trigger found by the PyCBC Live search using the exponential noise model ranking statistic (New Trigger) for the injection with index = 200. The ranking statistic value for both triggers is calculated from the signal rate and noise rate of the injections using Equation ??.

rate,  $\log p^N(\vec{\theta})$  and the ranking statistic and IFAR for both triggers calculated using the new statistic. The new trigger has a ranking statistic value of 3.12 which makes it the preferred trigger over the trigger found using the original statistic which has a ranking statistic value of -0.24. This increase in ranking statistic is caused almost entirely by the lower noise rate of the new trigger and while it also has a larger signal rate this is a marginal increase. We then compare this with Table ?? which

shows the components needed to calculate the original ranking statistic value and why the original trigger was chosen over the trigger preferred by the new statistic in the original search. We can see that the new trigger has lower  $\hat{\rho}$  in both detectors

Injection Index = 200	Original Trigger	New Trigger
$\rho_{new,H1}$	15.06	13.50
$\rho_{new,L1}$	6.67	6.58
$\rho_{new,H1}^2 + \rho_{new,L1}^2$	271.29	225.25
$\log p^S(\vec{\theta})$	-15.77	-15.67
$R_{old}$	15.48	13.93

Table 5.3: The H1 new SNR,  $\hat{\rho}_{H1}$ , L1 new SNR,  $\hat{\rho}_{L1}$ , squared sum of the H1 and L1  $\hat{\rho}$ s, (log) signal rate,  $\log p^S(\vec{\theta})$ , ranking statistic value,  $R_{old}$ , and inverse false-alarm rate (IFAR), calculated using the original ranking statistic used by PyCBC Live during the third observing run (Equation ??), for the preferred trigger found by the PyCBC Live search when using the original ranking statistic used during the third observing run and the different preferred trigger found by the PyCBC Live search using the exponential noise model ranking statistic for the injection with index = 200.

and a slightly higher signal rate. Therefore, the ranking statistic value of the original trigger is greater than that of the trigger preferred by the new ranking statistic.

We can do another analysis on the injection with index = 445 which was also found with different triggers in both searches. Table ?? shows the noise and signal rates for the injection where the noise rates are very similar but the new trigger's increase ranking statistic value is primarily due to an increase in signal rate with the new trigger's template. Similarly, when we look at the original statistic calculation in

Injection Index = 445	Original Trigger	New Trigger
$\log (\text{Noise Rate})$	-12.96	-12.69
$\log (\text{Signal Rate})$	-4.40	-2.74
$R_{new}$	8.56	9.95

Table 5.4: The (log) signal rate,  $\log p^S(\vec{\theta})$ , (log) noise rate,  $\log p^N(\vec{\theta})$ , ranking statistic value,  $R_{new}$  and inverse false-alarm rate (IFAR), calculated using the ranking statistic which includes an exponential noise model, for the preferred trigger found by the PyCBC Live search when using the original ranking statistic used during the third observing run (Original Trigger) and the different preferred trigger found by the PyCBC Live search using the exponential noise model ranking statistic (New Trigger) for the injection with index = 445. The ranking statistic value for both triggers is calculated from the signal rate and noise rate of the injections using Equation ??.

Table ?? we can see very similar original ranking statistic values where the signal rate



increase with the new trigger is not great enough to overcome the increased  $\hat{\rho}$  of the original triggers.

<b>Injection Index = 445</b>	<b>Original Trigger</b>	<b>New Trigger</b>
$\rho_{H1,new}$	10.44	10.78
$\rho_{L1,new}$	8.14	7.16
$\rho_{H1,new}^2 + \rho_{L1,new}^2$	175.25	167.47
log (Signal Rate)	-4.40	-2.74
$R_{old}$	12.90	12.73

Table 5.5: The H1 new SNR,  $\hat{\rho}_{H1}$ , L1 new SNR,  $\hat{\rho}_{L1}$ , squared sum of the H1 and L1  $\hat{\rho}$ , (log) signal rate,  $\log p^S(\vec{\theta})$ , ranking statistic value,  $R_{old}$ , and inverse false-alarm rate (IFAR), calculated using the original ranking statistic used by PyCBC Live during the third observing run (Equation ??), for the preferred trigger found by the PyCBC Live search when using the original ranking statistic used during the third observing run (Original Trigger) and the different preferred trigger found by the PyCBC Live search using the exponential noise model ranking statistic (New Trigger) for the injection with index = 445.

The addition of the exponential noise model in the ranking statistic can explain the increase in sensitivity for some injections, but others have an increase in sensitivity due to the different constructions of the ranking statistics. Different weighting has been placed on the signal and noise rates between the two searches and therefore a different trigger, with a better significance, can be chosen with a similar noise rate (the container for the exponential noise model) to the original trigger, but the template associated with that trigger has a better signal rate.

## 5.8 Investigating down-ranked regions

In sections ?? we investigated the effect of including the PSD variation in the ranking statistic and concluded that it will not enter the ranking statistic going forward. However, when looking at the sensitivity ratio for the new ranking statistic, including only the exponential noise model, against the original statistic (Figure ??) we can see that the decrease in sensitivity around at an FAR of 2 per month is still around 6%.

We have discussed how changing the ranking statistic can change the significance of individual injections in sections ??, ?? and, ?? and looking at the IFAR vs IFAR plot in Figure ?? we can highlight three regions of interest which contain down-ranked injections that we must investigate and understand the reasoning behind the down-ranking. The highlighted yellow region is of particular interest as being responsible for the decrease in sensitivity at a FAR of 2 per month.



### 5.8.1 Injections down-ranked around FAR = 2 per month

The sensitive volume ratio between the new ranking statistic search and the original ranking statistic search (Figure ??) highlights a drop in the sensitive ratio at an approximate FAR of 2 per month. The region responsible for this drop is highlighted on Figure ?? in a yellow dashed box and a zoomed in version of this figure can be seen in Figure ???. The original IFAR values of the injections found in this region range from 0.02 – 2.5 years and the new IFAR values range from 0.0015 – 0.0625 years, all these injections are found below the  $y = x$  diagonal line, indicating a lower IFAR found for the injections with the new ranking statistic. Figure ?? highlights four different parameters we can investigate to understand why these injections have been down-ranked. Injections that are found further from the diagonal  $y = x$  line have been down-ranked more significantly when using the exponential noise model in the ranking statistic. The 2 figures on the top row of Figure ?? show  $H_1 \alpha$  and  $H_1 \mu$  of the injections which represent the contribution to the ranking statistic from the exponential noise model. The majority of the injections have been found with a high noise rate, indicated by low  $\alpha$  and high  $\mu$  values, but there is no correlation between these values and the distance from the diagonal line.

The bottom right figure shows the combined log noise event rate density for the injections and again, there is no clear correlation between distance from the diagonal and noise rate value, the majority of injections have a high noise rate but some templates which have been heavily down-ranked have low noise rates relative to other injections in this region.

The bottom left figure shows the ratio of the two detector  $\hat{\rho}$  values identified in the search using the new statistic, calculated such that the ratio is always greater than 1. This figure shows that injections that have been most significantly down-ranked when using the new statistic have higher SNR ratios, this is caused by the noise rate contributions to the ranking statistic described in Section ?? and how the original statistic maximises the squared sum of the detector  $\hat{\rho}$  values in the noise rate calculation and the new statistic takes a linear sum of  $\hat{\rho}$  with a weighting based on historical noise rate of the trigger template. Therefore, in the new statistic search a low  $\hat{\rho}$  in one detector can no longer be compensated for by a much larger  $\hat{\rho}$  in the other detector, leading to a decrease in the significance of the injections. This is a consequence of the ranking statistic formulation. To solve this, the SNR ratios could be included in the noise rate calculations but, we have not analysed the potential negative effects this might cause.

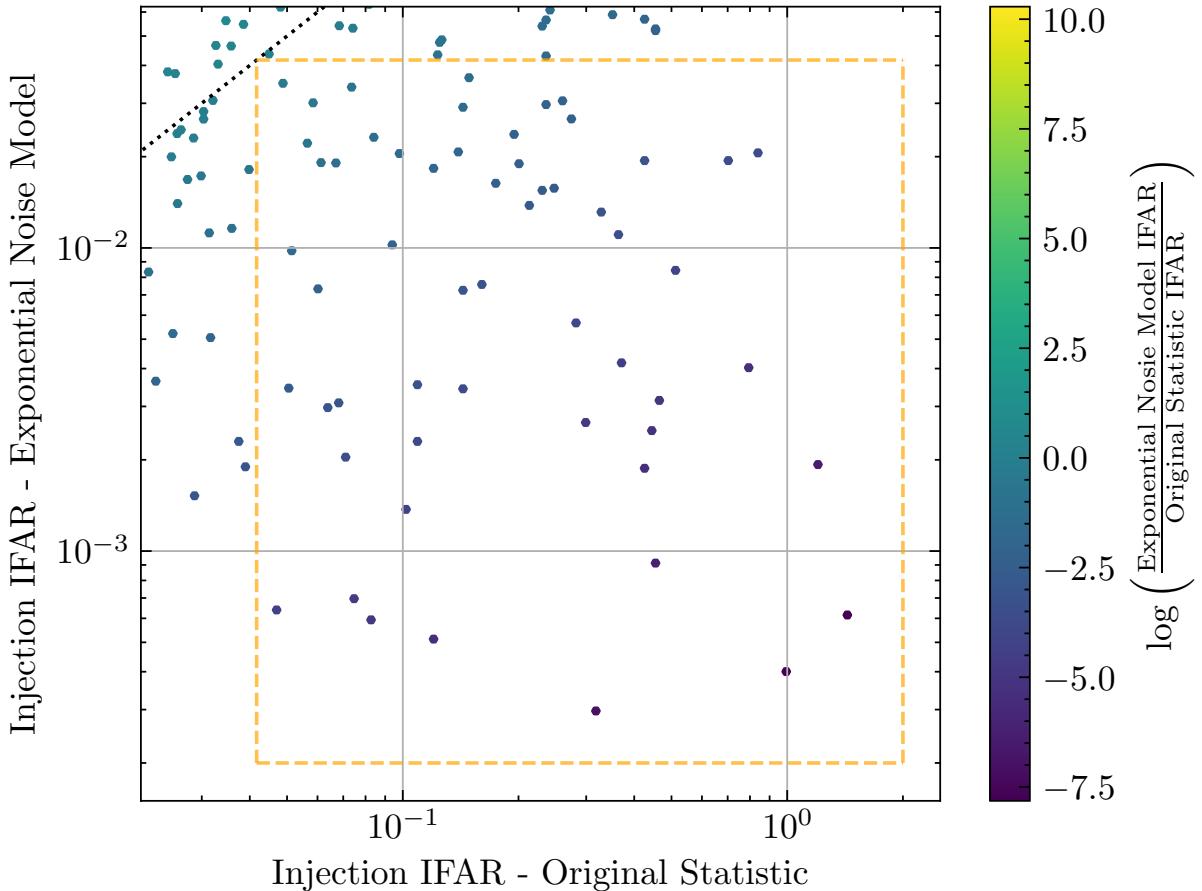


Figure 5.14: The inverse false-alarm rate (IFAR) values recorded for each injection seen by both the PyCBC Live search that includes the exponential noise model in the ranking statistic and the PyCBC Live search using the same ranking statistic used in the third observing run. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded using the original ranking statistic and the y-axis showing the IFAR recorded using the exponential noise model ranking statistic. The colour bar indicates the logarithmic difference in IFAR between the two searches, with positive values (yellow) indicating better performance when including the exponential noise model in the statistic and negative values (purple) indicating a better performance with the original statistic. The dashed line at  $y=x$  represents equal IFAR values for both searches. This figure is zooms in on the yellow dashed box shown in Figure ??.

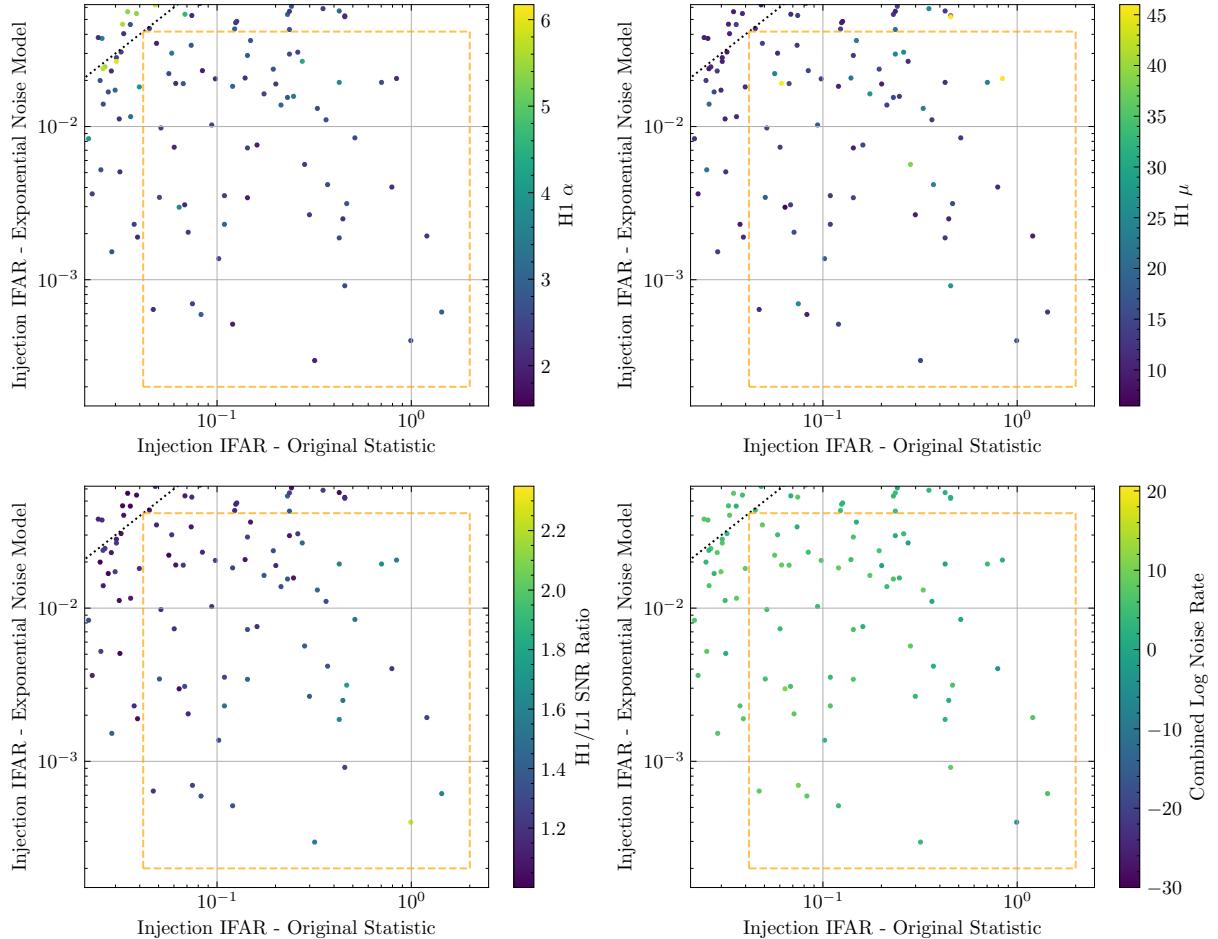


Figure 5.15: The inverse false-alarm rate (IFAR) values recorded for each injection seen by both the PyCBC Live search that includes the exponential noise model in the ranking statistic and the PyCBC Live search using the same ranking statistic used in the third observing run. Each point represents an injection from the injection set, with the x-axis showing the IFAR recorded using the original ranking statistic and the y-axis showing the IFAR recorded using the exponential noise model ranking statistic. The top left plot is coloured by the LIGO-Hanford trigger template's exponential fit factor,  $\alpha$ . The top right plot is coloured by the LIGO-Hanford trigger template's count of triggers found above  $\hat{\rho}_{threshold} = 6.0, \mu$ . The bottom left plot is coloured by the SNR ratio of the two detector  $\hat{\rho}$ , chosen such that the SNR ratio is always greater than 1. The bottom right plot is coloured by the combined (log) noise rate ??.



## 5.8.2 Down-ranked highly significant injections

The region indicated in the top right of Figure ?? by a dashed red box contains injections that were found with the maximum IFAR value with the original ranking statistic but have now been down-ranked by including the exponential noise rate model in the ranking statistic. These injections are still found with a high significance in the new statistic. This region contains 24 injections, which we can split into two groups for investigation: 12 injections found with the same trigger in both searches and 12 injections found with different triggers in both searches.

When an injection is found with the same trigger across both searches we can dismiss any changes in  $\hat{\rho}$  being the cause for an IFAR drop. We can analyse these injections to understand whether these injections were down-ranked due to high noise rate templates being responsible for the triggers, or if the construction of the ranking statistic has changed their significance.

For each injection found with the same trigger in both searches, Table ?? shows the individual detector  $\hat{\rho}$ , the template fit parameters  $\alpha$  and  $\mu$ , the single detector noise event rate density,  $r_n$ , the combined noise event rate density,  $p^N(\vec{\theta})$ , the signal event rate density,  $p^S(\vec{\theta})$ , and finally the ranking statistic value that these injections were found with using the new ranking statistic.

To demonstrate how the addition of the exponential noise model can cause the down-ranking of an injection, we pick injection 11608 which had the most significant drop, from maximum IFAR in the old statistic to an IFAR of 209.55 years when found with the new statistic. We display the ranking statistic components for both searches in tables ?? and ???. It is important to remember that we cannot directly compare ranking statistic values between the searches, the only values we can compare are the signal rate which is the same for both searches due to this injection being found with the same trigger in both, and the IFAR.

Looking at Table ?? for injection 11608 it has low  $\alpha$  values of 1.93 and 2.83 and therefore is being heavily down-ranked by the template fits, these amount to the highest noise rate in the table of  $-8.52$ . This injection has a signal rate of 1.07 which is low, the original statistic initially assigned such a large significance to this injection due to the high quadrature sum of the new SNRs heavily outweighing the low signal rate in the ranking statistic calculation (Equation ??) whereas the new statistic placed less significance on the  $\hat{\rho}$  from both detectors for this injection, with an equal contribution from both the signal and noise rate (Equation ??). While the exponential noise model has down-ranked the significance of this injection, it has still been found with a very high IFAR.



Index	H1				L1				$\log p^N(\vec{\theta})$	$\log p^S(\vec{\theta})$	Rank. Stat.
	$\hat{\rho}$	$\alpha$	$\mu$	$\log r_n$	$\hat{\rho}$	$\alpha$	$\mu$	$\log r_n$			
8015	8.53	2.60	24.5	-2.42	9.35	2.75	22.5	-5.07	-11.22	0.55	11.77
8026	9.31	2.37	9.4	-4.74	8.88	2.52	14.4	-3.67	-12.14	-0.91	11.23
8110	8.99	2.37	9.4	-3.97	10.30	2.52	14.4	-7.26	-14.96	-4.78	10.18
9222	10.09	2.78	15.0	-7.63	9.11	2.63	23.0	-4.07	-15.44	-3.04	12.40
9365	7.34	2.57	24.0	0.68	10.88	2.28	29.0	-6.94	-9.99	0.14	10.13
9777	8.33	2.57	33.75	-1.51	9.30	2.78	34.25	-4.63	-9.86	1.91	11.77
10105	9.23	2.01	13.67	-3.18	8.89	2.95	20.0	-4.46	-11.37	-0.31	11.06
10251	9.52	2.37	9.4	-5.23	10.26	2.52	14.4	-7.17	-16.14	-10.22	5.92
10979	8.31	2.01	13.67	-1.33	10.23	2.95	20.0	-8.41	-13.47	-2.02	11.45
11386	12.34	2.37	9.4	-11.91	6.47	2.52	14.4	2.41	-13.22	-5.02	8.20
11608	9.93	1.93	14.5	-4.24	7.68	2.83	23.75	-0.56	-8.52	1.07	9.59
13845	8.40	2.37	9.4	-2.57	10.37	2.52	14.4	-7.44	-13.74	-1.53	12.21

Table 5.6: Properties of preferred triggers detected by both the LIGO-Hanford (H1) and LIGO-Livingston (L1) detectors for injections indicated by index number in the injection set. The table shows the new signal-to-noise ratio ( $\hat{\rho}$ ), the exponential fit factor ( $\alpha$ ), the number of triggers above the threshold ( $\mu$ ), and the single detector (log) noise rate ( $\log r_n$ ) for the preferred trigger found by both detectors for each injection. The final columns provide the combined (log) noise rate ( $\log p^N(\vec{\theta})$ ), the (log) signal rate ( $\log p^S(\vec{\theta})$ ), and the ranking statistic value calculated using the ranking statistic including the exponential noise model.

Index = 11608 New Trigger	
$\log p^N$	-8.52
$\log p^S$	1.07
$R_{new}$	9.59
IFAR	209.55

Table 5.7

Index = 11608 Original Trigger	
$\rho_{H1,new}^2 + \rho_{L1,new}^2$	157.59
$\log p^S$	1.07
$R_{new}$	12.60
IFAR	30174.69

Table 5.8

When an injection has been found with a different trigger by the new ranking statistic and has still been down-ranked with a lower significance it indicates that the original trigger found by the original statistic would have performed even worse than the new trigger. In Section ?? we have discussed why different triggers are chosen by the new statistic and we can further demonstrate these using the injections in this region that have been found with a different trigger in the new ranking statistic search. We create the same table for the injections found with different template IDs, Table ???. These injections have very poor fit parameter values with low  $\alpha$  values (especially in H1) and high  $\mu$  values. One thing that is very clear, especially when compared to the same trigger table, is that these injections were all found with substantial SNR



Index	H1				L1				$\log p^N(\vec{\theta})$	$\log p^S(\vec{\theta})$	Rank. Stat.
	$\hat{\rho}$	$\alpha$	$\mu$	$\log r_n$	$\hat{\rho}$	$\alpha$	$\mu$	$\log r_n$			
8466	13.50	2.78	13.00	-17.27	6.58	3.01	17.50	2.21	-18.79	-15.67	3.12
8776	6.06	2.39	25.33	3.96	12.48	2.62	26.67	-12.74	-12.51	-0.22	12.29
9040	6.20	1.93	14.50	2.95	11.01	2.83	23.75	-9.99	-10.77	-1.30	9.47
9338	10.78	2.68	8.73	-9.66	7.16	2.08	10.96	0.70	-12.69	-2.74	9.95
9819	7.00	1.93	14.50	1.41	11.32	2.83	23.75	-10.87	-13.19	-1.25	11.94
9859	10.39	2.58	12.86	-7.83	6.61	3.69	14.57	1.74	-9.81	-4.62	5.19
10071	6.65	2.09	13.43	1.97	14.53	3.03	20.71	-21.69	-23.45	-16.34	7.11
10101	10.99	2.52	18.84	-8.72	7.65	3.87	17.00	-2.21	-14.66	-6.69	7.97
10134	6.55	2.60	24.50	2.72	8.76	2.75	22.50	-3.46	-4.47	0.09	4.56
10297	10.59	2.58	12.86	-8.35	7.96	3.69	14.57	-3.27	-5.35	-5.53	-0.18
10788	9.28	2.58	12.86	-4.95	8.38	3.69	14.57	-4.79	-13.47	-0.64	12.83
11505	10.78	2.78	13.00	-9.71	6.67	3.01	17.50	1.95	-11.48	-5.97	5.51

Table 5.9: Properties of preferred triggers detected by both the LIGO-Hanford (H1) and LIGO-Livingston (L1) detectors for injections indicated by index number in the injection set. The table shows the new signal-to-noise ratio ( $\hat{\rho}$ ), the exponential fit factor ( $\alpha$ ), the number of triggers above the threshold ( $\mu$ ), and the single detector (log) noise rate ( $\log r_n$ ) for the preferred trigger found by both detectors for each injection. The final columns provide the combined (log) noise rate ( $\log p^N$ ), the (log) signal rate ( $\log p^S$ ), and the ranking statistic calculated using the ranking statistic including the exponential noise model.

ratios. These injections and their SNR ratios can be seen in Figure ?? and this is a clear demonstration of how high SNR ratio triggers are favoured in the old statistic and are not in the new statistic. Overall, most injections have been found with high combined detector noise rates and the signal rates aren't large enough to compensate. This region demonstrates that while injections may be down-ranked using the new ranking statistic we are still achieving an overall increase in sensitivity in this IFAR region (Figure ??) and we understand the causes of the down-ranking of these highly significant injections.

### 5.8.3 Down-ranked marginal injections

The injections found in the blue dashed box in the middle of Figure ?? are injections which have been found with IFAR above  $\sim 1$  year with the original ranking statistic but have been down-ranked by the new ranking statistic. An IFAR of 1 year is a conservative threshold that distinguishes gravitational-wave events as being real or fake, the sensitive ratio plot (Figure ??) highlights the sensitivity increase at this IFAR threshold as being 43%. These are the few injections in this region that have been down-ranked and so we want to understand what has caused this down-ranking.



This region only contains 3 injections and therefore we can investigate each injection individually to understand the cause of their down-ranking.

We can look at the numerical values associated with each injection, this is shown in Table ??.

<b>Index</b>	<b>H1</b>				<b>L1</b>				<b>New</b>			<b>Old</b>		
	$\hat{\rho}$	$\alpha$	$\mu$	$\log r_n$	$\hat{\rho}$	$\alpha$	$\mu$	$\log r_n$	$\log p^N$	$\log p^S$	IFAR	$\hat{\rho}_{H1}^2 + \hat{\rho}_{L1}^2$	$\log p^S$	IFAR
8313	5.87	2.39	15.50	5.33	12.21	3.28	19.75	-16.20	-14.60	-15.17	0.57	183.43	-15.17	40.18
10256	9.57	2.71	16.16	-5.92	8.04	3.77	19.30	8.95	-0.69	-1.65	0.35	119.74	-1.65	1.24
10995	8.73	2.32	10.00	-2.05	10.07	1.44	18.21	-1.26	-7.04	-8.38	0.28	151.36	-8.38	3.06

Table 5.10: Properties of preferred triggers detected by both the LIGO-Hanford (H1) and LIGO-Livingston (L1) detectors for injections indicated by index number in the injection set. The table shows the new signal-to-noise ratio ( $\hat{\rho}$ ), the exponential fit factor ( $\alpha$ ), the number of triggers above the threshold ( $\mu$ ), and the single detector (log) noise rate ( $\log r_n$ ) for the preferred trigger found by both detectors for each injection. The ‘New’ columns provide the combined (log) noise rate ( $\log p^N$ ), the (log) signal rate ( $\log p^S$ ), and the inverse false-alarm rate (IFAR) calculated using the ranking statistic including the exponential noise model. The ‘Old’ columns provide the squared sum of the two detector  $\hat{\rho}$ s, the (log) signal rate ( $\log p^S$ ), and the IFAR values calculated using the ranking statistic used by the PyCBC Live search during the third observing run.

---

All three of these injections have been found with the same trigger in both searches and demonstrate a few different cases for injection down-ranking. Injection 8313 was found with the same trigger in both searches and while the new statistic's combined noise rate is very low it isn't enough to overcome the very low signal rate whereas the original statistic placed more weight on the very high L1  $\hat{\rho}$  which has compensated for the relatively low H1  $\hat{\rho}$ . This is another case where the SNR ratio has a big impact on the significance of recovered events in the original statistic search. Injection 10256 was originally found with an IFAR of 1.24 and has been down-ranked to an IFAR of 0.35. The L1 noise rate is very high, contributing to a large combined noise rate which, when matched with the signal rate, has not produced a very significant ranking statistic value. Finally, injection 10955 is seen with decent  $\hat{\rho}$  in both detectors, but the  $\alpha$  values are poor, especially in L1. This produces a high combined log noise rate which the signal rate couldn't overcome.

## 5.9 Testing using the full template bank

To test the implementation of the new additions to the ranking statistic we ran the PyCBC Live search in an ‘offline’ mode where the PyCBC Live search operates how it will for the real low-latency search but all the data is available before the search begins and there is no need to wait for the next analysis segment. This allows the PyCBC Live search to process the data as fast as possible and when using the PyCBC-BBH template bank we could process one week of gravitational-wave data in as little as a day.

The PyCBC-BBH template bank contains  $\sim$ 15,000 templates, whereas the template bank used by the PyCBC Live search during the fourth observing run has over 700,000 templates. Individual users are limited to the amount of memory that can be requested when generating the template bank and so the changes could not be tested using the full template bank during development. To fully test these changes, we must use the full template bank and a live dataset which simulates exactly how the real search will operate.

The gravitational-wave mock data challenge (MDC) replays data from the third observing run with a known set of injections and streams it to the same data infrastructure which streams live data from the gravitational-wave detectors. This allows the full simulation of the low-latency gravitational-wave search and unlocks the ability to test changes to the gravitational-wave low-latency search pipelines even when the gravitational-wave detectors are not operating, either between observing runs, during maintenance downtime or when the detectors are not working correctly.



We were able to test our changes to the PyCBC Live ranking statistic on the PyCBC Live MDC computational resources, allowing the new ranking statistic to be run a low-latency environment with live-streamed data to test for any latency introduced into the search by the new ranking statistic components. The computational resources assigned to the PyCBC Live MDC search are equal to those being used in the PyCBC Live search during the fourth observing run and therefore we do not encounter the previously mentioned memory allocation limitations and related issues.

To test the ranking statistic changes, we created static template fits for all templates in the full template bank for a week of data and ran the search on the proceeding weeks during the test. The template fits were not updated during this test and the purpose of the test was not to measure sensitivity improvements but to test for bugs in implementation and if the additional ranking statistic components introduce any lag to the search. During this test we identified a number of bugs which were introducing major lag to the search, after fixing the bugs there is no latency difference between the configuration used during the third observing run and the PyCBC Live search including the new ranking statistic components. Figure ?? shows the IFAR for gravitational-wave candidates identified by both the original ranking statistic and the exponential noise model ranking statistic for a single day during the MDC test.

## 5.10 Conclusion

We have demonstrated that two ranking statistic components taken from the PyCBC Offline search’s ranking statistic can be adapted and used in the PyCBC Live search’s ranking statistic to improve the overall sensitivity of the PyCBC Live search for gravitational waves. We have implemented the new ranking statistic additions, PSD variation and an exponential noise model, into the PyCBC Live ranking statistic and searched through 1 week of gravitational-wave data containing thousands of gravitational-wave injections with a PyCBC Live search using the original ranking statistic used by PyCBC Live during the third observing run and another PyCBC Live search using a ranking statistic that includes PSD variation and the exponential noise model and compared the false-alarm rates of the injections found by both searches to measure the increase in sensitivity from including the new additions to the ranking statistic.

We measure an overall increase in sensitivity when both PSD variation and the exponential noise model are included in the ranking statistic however, we measure a greater increase in sensitivity when we do **not** include PSD variation in the ranking statistic. We also investigate a decrease in sensitivity around a false-alarm rate of 2

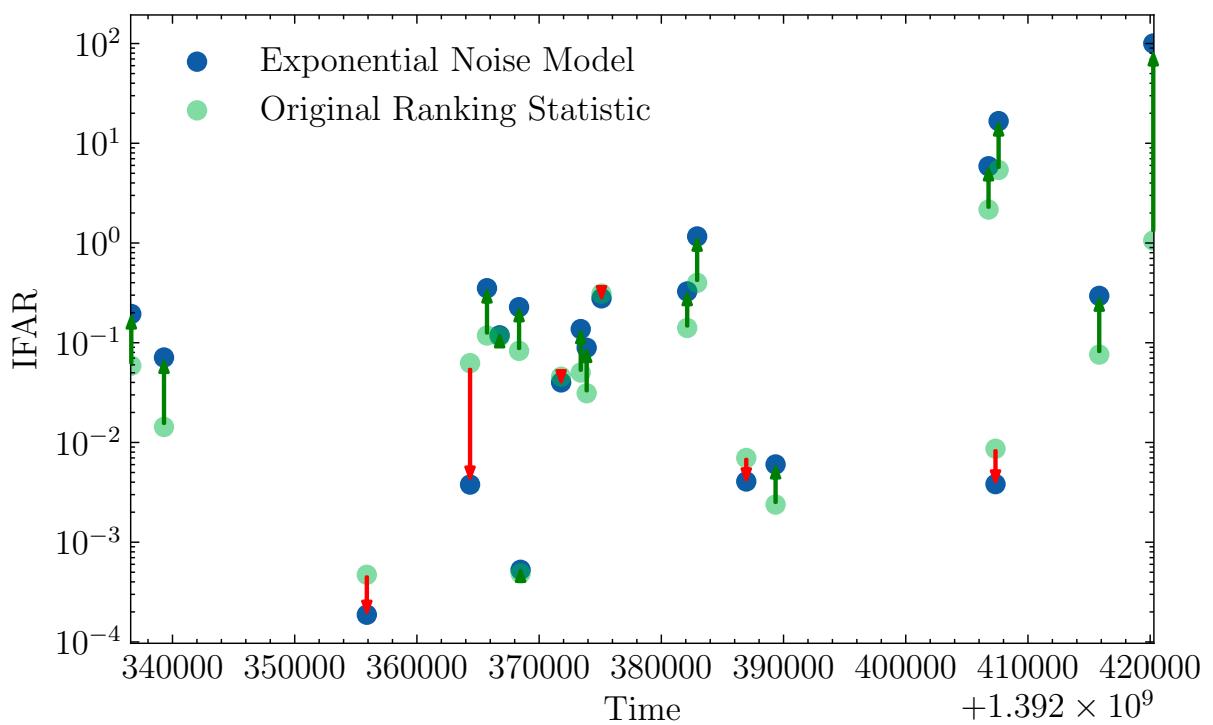


Figure 5.16: Gravitational-wave candidates jointly seen by the PyCBC Live full bandwidth Mock Data Challenge (MDC) searches using the original PyCBC Live third observing run ranking statistic (Original Ranking Statistic, green circles) and the PyCBC Live ranking statistic including the exponential noise model (Exponential Noise Model, blue circles). Events which have been seen with an increased inverse false-alarm rate with the exponential noise model ranking statistic have a green arrow between the original search's circle and the exponential noise model's circle, events seen with a decreased inverse false-alarm rate have a red arrow.



per month by identifying the changes the exponential noise model has made to the ranking statistic to cause the down-ranking of the injections originally found with this false-alarm rate.

We have tested the changes to the PyCBC Live search using the smaller PyCBC-BBH template bank and the complete PyCBC Live template bank being currently used in the fourth observing run and we highlight the difficulty of increasing the complexity of the low-latency search pipeline ranking statistic where changes have the potential to introduce latency to the search pipeline, delaying the dissemination of search results and how the new additions to ranking statistic have not increased the analysis latency of the PyCBC Live search.

The new additions to the PyCBC Live ranking statistic are currently being tested alongside PyCBC Live infrastructure changes to allow template fit statistics to update weekly based on the previous week's triggers to adapt to changing template noise distributions. The new PyCBC Live ranking statistic will also include data quality information from iDQ and will be run on the second half of the fourth observing run.

# 6

---

## Evaluating the PyCBC Live Early Warning Search

---

*“Stars are better off without us.”*

— *Josephus Miller, Leviathan Wakes*

The work that forms this chapter was performed as my final project during the PhD to investigate and improve the PyCBC Live Early Warning search. This work has highlighted a few problems with the search and has identified the solutions to these problems, if more time were available the next steps would be to implement these solutions into PyCBC Live. These solutions will be implemented by other researchers at the University of Portsmouth. The PyCBC Live early warning search identifies gravitational-wave signals prior to merger, giving warning to the international astronomy community of an incoming gravitational-wave event. The early warning search is an adaptation to the PyCBC Live full bandwidth search—discussed heavily in chapter ??—and we have identified problems relating to search for pre-merger signals which require changes to be made to the early warning infrastructure to improve the sensitivity of the pipeline.

In Section ?? we discuss the joint detection of astrophysical events with both gravitational wave and electromagnetic observatories, in Section ?? we highlight the only multi-messenger event seen thus far and the timeline of observations. The PyCBC Live early warning search is described in Section ?? with the template bank construction



in Section ?? and how sky localisation for events is obtained in Section ?? . Section ?? uses GW1701817 as an example to demonstrate the capabilities of the early warning search and in Section ?? we describe a full injection set test performed using the early warning search to identify any problems and the results of this search. Finally, in sections ?? and ?? we discuss the problems found when testing the early warning search, the cause of the problems and the potential solutions to these problems.

## 6.1 Multi-messenger astronomy

The PyCBC gravitational-wave search pipeline searches for gravitational-wave signals from the merger of two compact objects [? ]. Black holes do not emit electromagnetic radiation and therefore the merger of two black holes can currently only be directly observed with gravitational-wave detectors [? ]. Neutron stars, on the other hand, are electromagnetically bright themselves and the merger of two neutron stars might produce a kilonova [? ]. Kilonovae emit a broad range of electromagnetic radiation across the spectrum [? ] which can be observed with many of the ground and space-based electromagnetic observatories we have on Earth. The combination of gravitational wave and electromagnetic emissions allows us to understand more about these events than either observation could provide alone [? ]. Astrophysical events which have been observed with more than one type of signal are called multi-messenger events, this can be from any of the three: electromagnetic signals, gravitational waves or neutrinos. We will use the term ‘multi-messenger event’ to describe an event which has been seen with a gravitational-wave signal and an electromagnetic counterpart.

One such quantity that can be measured from observing kilonovae with both gravitational waves and electromagnetic radiation is the Hubble constant [? ], the measured rate of expansion of the Universe [? ]. The Hubble constant is currently determined across two distance scales using purely electromagnetic observations: firstly, the large scale cosmological measurements using the cosmic microwave background [? ] and baryon acoustic oscillations [? ] and secondly, the local Universe scale using the astrophysical standard candle Cepheid variable stars [? ] and type Ia supernovae [? ]. The Hubble constant values obtained at the different scales do not agree [? ] so by combining the observation of binary neutron star signals using both gravitational wave and electromagnetic observatories we can provide an independent measurement of the Hubble constant to break this tension.

At nearby distances ( $d \lesssim 50\text{ Mpc}$ ) the Hubble constant can be calculated using Hubble’s law [? ],

$$v = H_0 d, \quad (6.1)$$

where  $v$  is the recessional velocity of a source,  $H_0$  is the Hubble constant and  $d$  is the proper distance to the source. For nearby objects ( $z \lesssim 0.1$ ):  $v$  can be approximated as  $v \approx cz$ , where  $z$  is the redshift of the object and  $c$  is the speed of light; luminosity distance is approximately equal to the proper distance  $d \approx D_L$ . On small distance scales the expansion of the universe doesn't greatly affect these measurements. We obtain an equation for the Hubble constant at short distances

$$H_0 = c \frac{z}{D_L}. \quad (6.2)$$

The expansion rate of the Universe historically is not constant. For objects at larger distances we have to account for redshift in the luminosity distance. We begin by defining the luminosity distance in terms of the comoving distance,

$$D_L(z) = (1 + z)D_c(z), \quad (6.3)$$

which accounts for the expanding universe in the term  $(1 + z)$ . The comoving distance is given by

$$D_c(z) = c \int_0^z \frac{dz'}{H(z')}, \quad (6.4)$$

where the Hubble constant,  $H(z')$ , is now the Hubble parameter at redshift  $z'$  and depends on the model of cosmology. We can use the comoving distance to get an expression for the luminosity distance,

$$D_L(z) = (1 + z)c \int_0^z \frac{dz'}{H(z')}. \quad (6.5)$$

$H_0$  will depend on the cosmological parameters representing the density of matter, radiation and dark energy. To measure the Hubble constant we need to measure the redshift and luminosity distance of multiple sources.

The luminosity distance of a gravitational-wave event is inversely proportional to the amplitude of the gravitational-wave strain [? ],

$$h(t) \propto \frac{1}{D_L}, \quad (6.6)$$

which is obtained via gravitational-wave searches and parameter estimation. The gravitational-wave strain amplitude does have a degeneracy with the inclination angle,  $\iota$ , of the binary neutron star system which is described as the angle between the line of sight to the observer and the orbital angular momentum of the source [? ].

The gravitational-wave strain dependence on inclination angle manifests differ-



ently in the two gravitational-wave polarisations and can be simply described as [? ].

$$h_+ \propto (1 + \cos^2 \iota), \quad (6.7)$$

$$h_\times \propto \cos \iota. \quad (6.8)$$

It can be seen from Equation ?? that an edge-on system ( $\iota = 90^\circ$ ) will produce a signal entirely in the  $h_+$  polarisation but a face-on system will produce a mixed signal with contributions from both  $h_+$  and  $h_\times$ . The inclination angle degeneracy could be broken if we were able to measure  $h_+$  and  $h_\times$  independently, but the individual gravitational-wave detectors are sensitive only to a linear combination of the polarisations [? ]. Multiple gravitational-wave detectors with different orientations will be sensitive to different linear combinations of  $h_+$  and  $h_\times$  and are therefore required to break this degeneracy and fully reconstruct the source orientation [? ].

The redshift of a gravitational-wave event is embedded in the gravitational-wave signal via the chirp rate of the system. The chirp rate is defined as the rate of change of the gravitational-wave frequency and is given to leading order as [? ]

$$\frac{df}{dt} = \frac{96}{5} \pi^{8/3} \left( \frac{G\mathcal{M}}{c^3} \right)^{5/3} f^{11/3}. \quad (6.9)$$

The rate of change of the gravitational-wave frequency is easy to measure and will give us the chirp mass of the system,

$$\mathcal{M} = \frac{(m_1 m_2)^{\frac{3}{5}}}{(m_1 + m_2)^{\frac{1}{5}}}. \quad (6.10)$$

which depends on the two masses of the system,  $m_1$  and  $m_2$ . The observed  $\mathcal{M}$  is redshift dependent,

$$\mathcal{M}_{obs} = \mathcal{M}_{source}(1+z), \quad (6.11)$$

and  $\mathcal{M}_{obs}$  is scaled by the redshift where events with larger redshifts will appear with larger observed chirp masses than their source chirp mass,  $\mathcal{M}_{source}$ . We would require a measurement of the Hubble constant to determine the redshift using the luminosity distance and we reach a circular argument.

We can look at other ways to measure the redshift of the event without using the gravitational-wave signal. The primary candidate is an electromagnetic emission from the event, for example, a kilonova from a binary neutron star system. Using the electromagnetic emission we can locate the host galaxy of the source and find the

---



approximate redshift of the binary neutron star system. We will then have a luminosity distance, measured by the gravitational-wave signal, and a redshift, measured by the electromagnetic observation, and using Equation ?? we can calculate the Hubble constant. This combination of event information from two separate sources of information for the same event means we can use electromagnetically bright gravitational-wave sources as another standard siren for measuring the Hubble constant, highlighting the importance of observing multi-messenger events.

## 6.2 GW170817: The only multi-messenger event so far

We have described multi-messenger events as those that have been seen with a gravitational wave and an electromagnetic signal, in reality the event will broadcast information across the electromagnetic spectrum at varying timescales so we might see a single gravitational-wave signal but multiple electromagnetic signals. There are observatories around the globe and in space which span this entire spectrum and can observe counterparts to our gravitational-wave signal in all the different frequency ranges: from high frequency gamma rays to long wavelength radio waves.

GW170817 [? ] is the first and only multi-messenger event we have observed, the Q-scans depicting the gravitational-wave signal found by the three detectors online at the time of the event can be seen in Figure ?? and the GCN circular [? ] (rapid astronomical bulletins submitted by and distributed to the global physics community) can be found at: <https://gcn.gsfc.nasa.gov/other/G298048.gcn3>. On the 17th August 2017 at precisely 12:41:04 UTC, the gravitational waves from the merger of a binary neutron star system were seen by LVK detectors. Only 1.7 seconds later, a gamma-ray burst (GRB 170817A [? ]) was seen by Fermi [? ? ] and INTEGRAL [? ? ]. The kilonova emissions were seen next, visual light by the Hubble Space Telescope (HST) [? ? ] at 11 hours, infrared by HST, VISTA [? ? ] and Spitzer [? ? ] at 12 hours and UV by Swift’s UVOT [? ? ] at 15.3 hours post-merger. The final electromagnetic frequencies to be seen were X-rays by Chandra [? ] at 9 days post-merger and radio waves by VLA [? ? ] at 16 days post-merger.

Analysing GW170817’s gravitational-wave signal and electromagnetic emissions reveal a binary neutron star system located in the Hydra constellation belonging to the galaxy NGC 4993 [? ]. Using these signals the Hubble constant can be inferred as  $H_0 = 70.0^{+12.0}_{-8.0} \text{ km s}^{-1} \text{ Mpc}^{-1}$  [? ] which is consistent with both the measured cosmological value for  $H_0$  ( $67.74 \pm 0.46 \text{ km s}^{-1} \text{ Mpc}^{-1}$  [? ]) and the astrophysical value for  $H_0$  ( $73.24 \pm 1.74 \text{ km s}^{-1} \text{ Mpc}^{-1}$  [? ]) but, it is not accurate enough to break the tension between both values. To constrain the value of the Hubble constant further, we

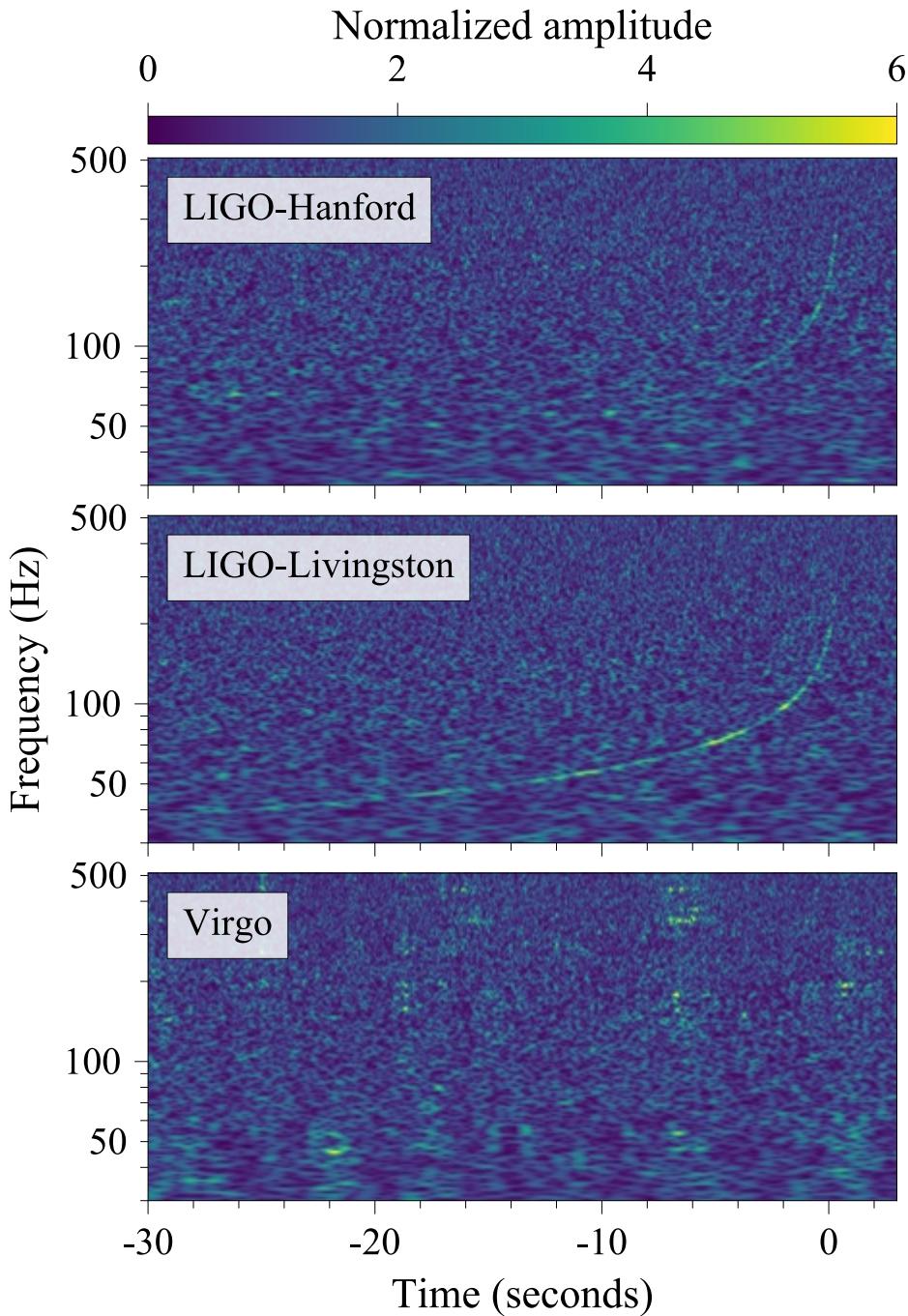


Figure 6.1: The three detector (LIGO-Hanford (top), LIGO-Livingston (middle) and Virgo (bottom)) time-frequency images (Q-scan [? ]) showing data containing the first binary neutron star gravitational-wave event, GW170817 [? ], with times shown relative to 17th August 2017 12:41:04 UTC. The detector amplitudes are individually normalised for that detector’s amplitude spectral density. The glitch observed in the LIGO-Livingston detector has been subtracted, the technique and results when doing so are presented in section IV of [? ] and this figure is taken directly from figure 1 in [? ].

---

need to observe many more electromagnetically bright events [? ]. Using GW170817 we were able to determine other physical properties such as the speed of gravity [? ? ].

Using the electromagnetically bright multi-messenger events to constrain the measurement of the Hubble constant is straight forward. We are also able to use ‘dark siren’ events, which have no electromagnetic counterpart and so rely entirely on the gravitational-wave emission to measure the redshift. To identify the redshift of dark sirens must use galaxy catalogues and gravitational-wave sky maps to locate the host galaxy of the gravitational-wave signal. This is not discussed further in this chapter, for more information on dark sirens please look to [? ? ]

## 6.3 Detecting gravitational-wave events pre-merger

Gravitational-wave signals from low-mass sources merge at frequencies above the sensitive frequency band of the LIGO gravitational-wave detectors [? ]. The lower sensitive frequency of the LIGO detectors is  $\sim 20$  Hz meaning the gravitational-wave signal can exist in the sensitive band for potentially hundreds of seconds before it is identified by the gravitational-wave search pipelines. Figure ?? shows the accumulation of SNR over a  $1.4 M_{\odot}$ – $1.4 M_{\odot}$  binary neutron star signal; at 29 Hz, 31 % of the SNR is accumulated at 83 Hz of the total template duration.

This presents an opportunity to detect the gravitational-wave signal before it merges, in the inspiral regime of the system. The visible electromagnetic counterpart to the gravitational-wave signal is produced post-merger, therefore, by disseminating information about an ongoing gravitational-wave event prior to the merger we are able to give enough warning to electromagnetic observatories to slew their telescopes to the approximate region on the sky in which we will indicate a gravitational-wave event will occur. Providing this warning pre-merger is commonly referred to as ‘early warning’. This section details the methodologies used in the PyCBC Live early warning search to achieve the detection of electromagnetically bright gravitational-wave signals pre-merger.

Even with an accurate sky map and dedicated electromagnetic telescopes (for example, GOTO is used for gravitational-wave event follow-up [? ]), the gamma-ray burst counterparts can arrive less than two seconds after merger, not giving us enough time to identify the signal, produce a sky map and slew any telescopes to the correct sky location.

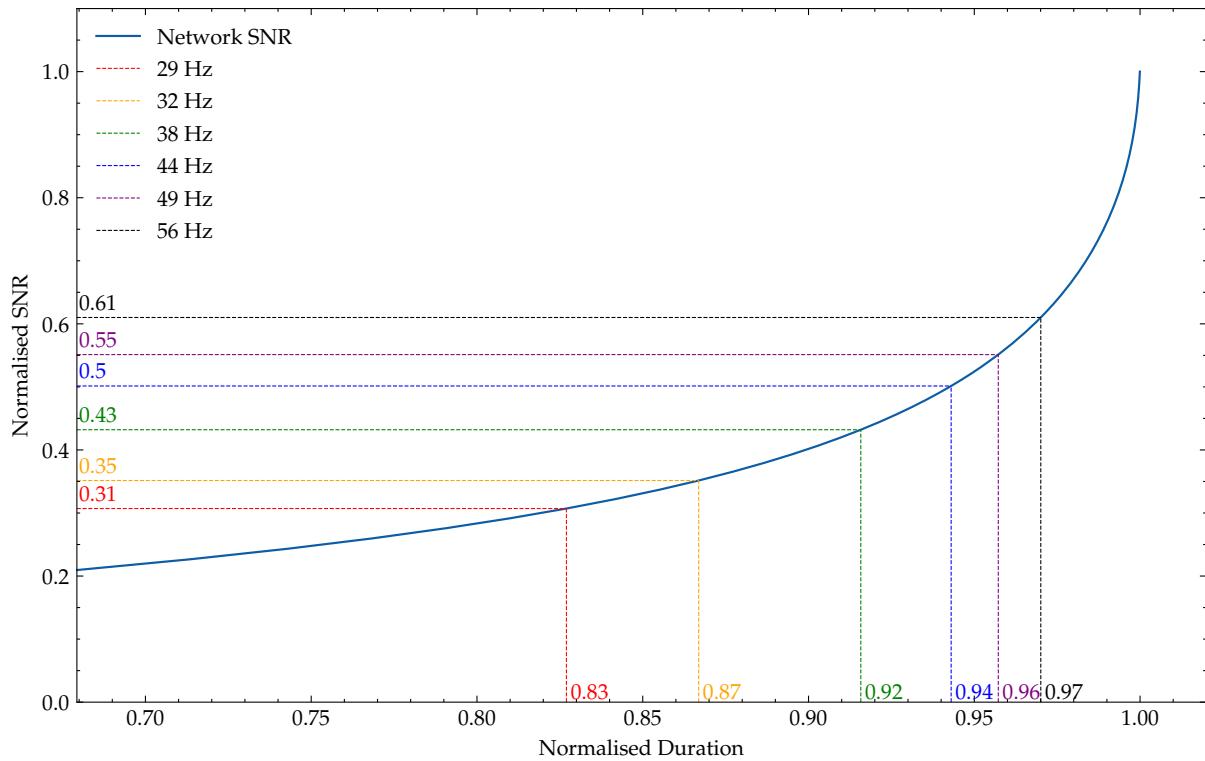


Figure 6.2: The accumulation of signal-to-noise ratio over the duration of a  $1.4 M_{\odot}$ – $1.4 M_{\odot}$  binary neutron star system. SNR and duration have been normalised such that the SNR and duration at an initial signal frequency of 15 Hz and a final signal frequency of 1024 Hz is equal to 1.



### 6.3.1 The PyCBC Live early warning search

To identify potential gravitational-wave signals prior to the merger of the compact objects, the PyCBC Live search pipeline has created an optimised early warning search for pre-merger gravitational waveforms. In this section, we describe the design choices which allow the early warning search to operate in the early warning regime.

We use two terms when referring to search detection times: latency refers to the time taken by the search to identify a gravitational-wave template and produce a gravitational-wave event; warning refers to the time prior to the gravitational-wave signal’s merger that the event was identified. The early warning attempts to maximise the warning time, which is improved by decreasing the latency between detection and event production.

We can divide the latency of detection into three contributing components: firstly, the analysis stride of the early warning search is one second, therefore a signal will have to wait until the entire second has arrived before it can be processed; secondly, when highpassing and overwhitening<sup>1</sup> our data we introduce corruption to the edge samples, we pad the data by an extra 1.5 seconds to account for this, which the signal has to wait for; thirdly, we need to analyse the data to detect the signal, we impose a strict analysis time requirement of less than the analysis stride, so data does not build up in the processing queue. In total, we have a minimum latency of 2.5 seconds (if the signal appears in the final few samples of the analysis stride) and a maximum latency of 3.5 seconds (if the signal appears in the first few samples of the analysis stride). Figure ?? shows minimum and maximum latencies of both the full bandwidth and early warning search.

---

<sup>1</sup>Dividing the data by the PSD [? ].

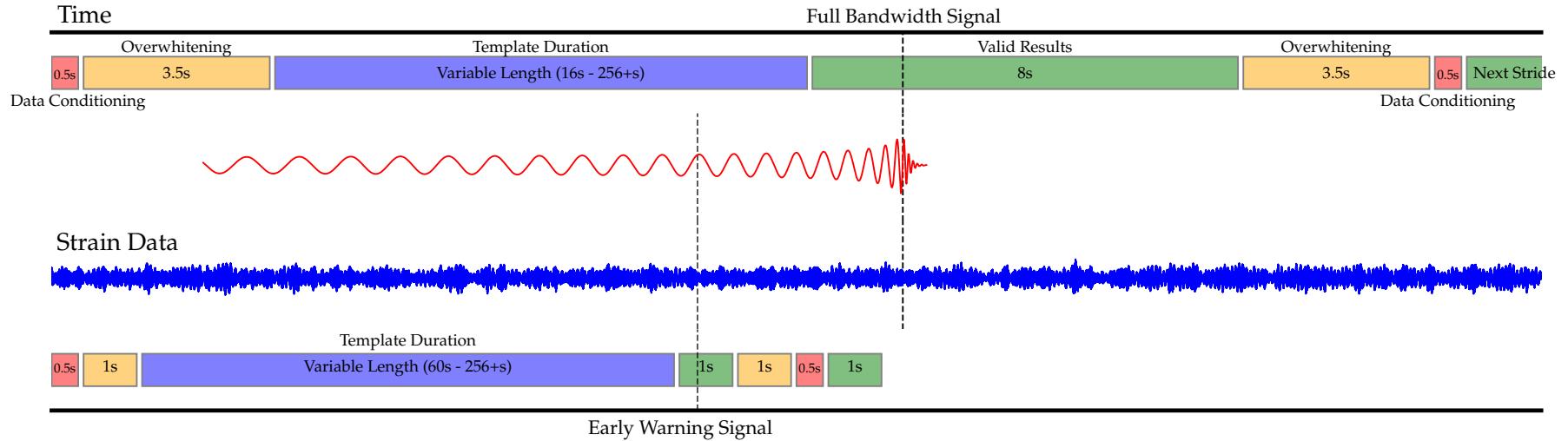


Figure 6.3: An illustration of the detection latency between template end and event upload to GraceDB [? ]. The top half of the plot pertains to the PyCBC Live Full Bandwidth search and indicates that the search has a maximum latency of 20 seconds, the bottom half describes the PyCBC Live Early Warning search which has a maximum latency of 3.5 seconds. From the end of the gravitational-wave template the search must wait for the current analysis stride to end, another period of data must then be waited on (overwhitening and data conditioning) and then the search analyses the previous analysis stride during the next analysis stride.

---

By comparison, the full bandwidth search has a minimum and maximum latency of 12 to 20 seconds respectively, with no possibility to observe the event pre-merger.

To ensure the processing time for the triggers is completed in less than one second aspects of the search have been simplified: we search over only two interferometers (LIGO-Hanford and LIGO-Livingston) and we use a simple new SNR single detector ranking statistic and phase-time-amplitude histogram coincident ranking statistic. Alongside this, the early warning search has a far smaller template bank of only 9180 templates compared to the  $\sim 700,000$  templates in the full bandwidth template bank.

### 6.3.2 Frequency truncated template bank

The template bank used by the PyCBC Live early warning search is composed of frequency truncated templates. These are gravitational-wave templates which start at a lower frequency and end in the inspiral phase at a upper frequency cutoff. The bank contains six different frequency cutoffs at: 29, 32, 38, 44, 49 and 56 Hz. These frequencies correspond to approximately 60, 46, 29, 20, 15 and 10 seconds before merger for a  $1.4 M_{\odot}$ - $1.4 M_{\odot}$  binary neutron star signal.

The PyCBC Live early warning template bank is constructed by generating six separate template banks (one for each frequency cutoff) using a geometric placement algorithm [? ] with a minimal match between templates of 0.97. The template parameters of the bank are chosen to represent potentially electromagnetically bright signals, these are produced by the merger of low mass binary neutron star or neutron star black hole systems. The masses of the two compact objects is limited between  $1\text{--}3 M_{\odot}$  and to reduce the size and complexity of the template bank we use zero-spin on both components. The PSD used to generate the template bank is representative of the Advanced LIGO design curve that includes estimates for the main fundamental noises of the interferometer, in particular: seismic noise, thermal noise and quantum noise [? ] with a binary neutron star range of 140 Mpc [? ]. The six early warning template banks combined contain 9180 templates. The duration of templates at each final frequency cutoff can be seen in Figure ?? as a function of the chirp mass,  $M$ , of the template. It can be seen that higher  $M$  templates have shorter duration and templates with a lower frequency cutoff also have shorter duration when compared to templates with the same  $M$  but a higher frequency cutoff.

### 6.3.3 Determining event sky locations

We call a gravitational-wave candidate obtained by the early warning search an ‘event’. These events have information about the gravitational-wave parameters and make a

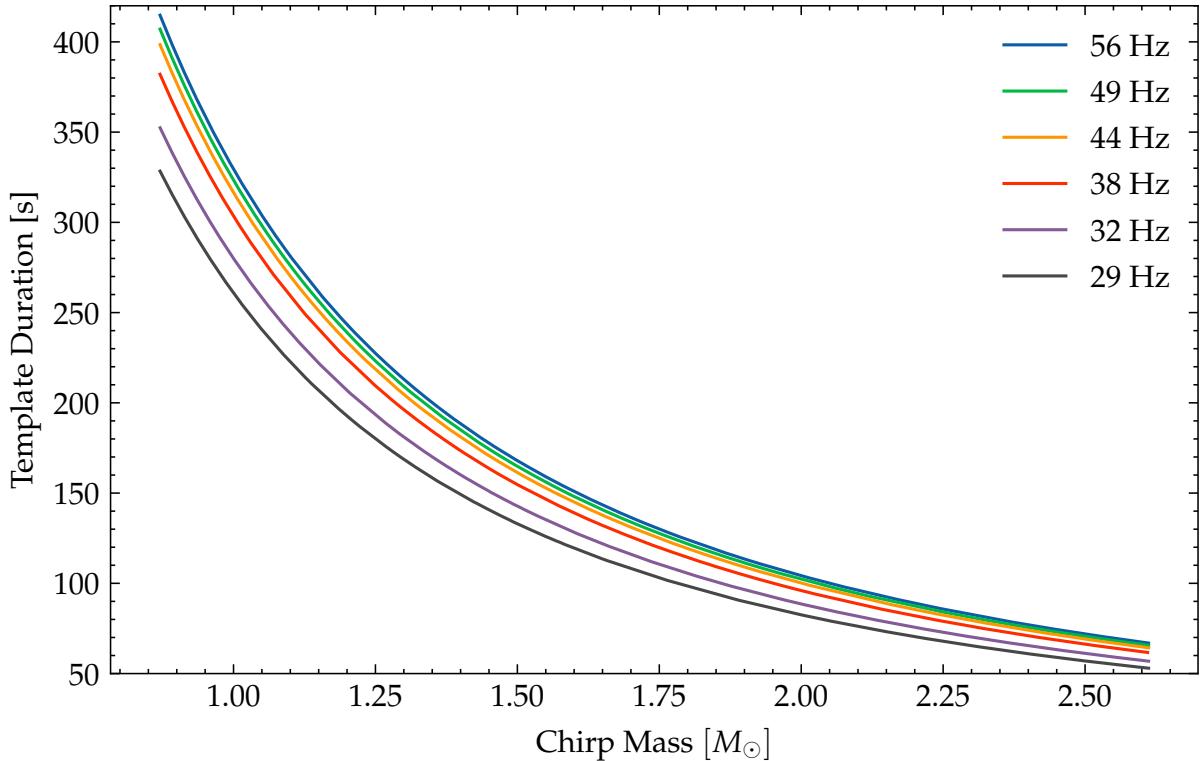


Figure 6.4: Template duration versus chirp mass,  $\mathcal{M}$ , for different frequency cutoffs in the early warning template bank. Each curve corresponding to a different frequency cutoff,  $f_{final}$ , (29–56 Hz). The chirp mass is calculated from the component masses of the binary system, and the template duration is calculated between the lower frequency, 17 Hz and the frequency cutoff,  $f_{final}$  Hz.

---

prediction of the time of coalescence of the gravitational-wave signal. Identifying the sky location of the event is critical for multi-messenger astronomy. With a sky location, the gravitational-wave observatories are able to tell electromagnetic observatories where to slew their telescopes to see the event prior to merger, to have the best chance of seeing electromagnetic counterparts as possible.

We describe the sky location with three parameters: right ascension, declination, and distance. Distance is measured from the gravitational-wave signal, as previously described and we measure the right ascension and declination using two key pieces of information: the time difference between time of arrivals at different detectors which allows us to calculate a time-of-arrival triangulation; and amplitude differences between the signal strength measured by each detector to determine the gravitational-wave polarisation, the individual detector quadrupolar sensitivity pattern of the antenna response will favour particular sky locations where the detector is more sensitive to gravitational waves coming from certain angles. The greater number of gravitational-wave detectors in the network lead to more accurate sky location triangulation. Figure ?? shows the sky map created for GW170817 [?] with the source location identified by the LIGO-Hanford and LIGO-Livingston detectors shown in the green dashed contour (representing the 90% confidence interval).

A greater number of detectors observing the event will produce more accurate sky locations; adding a third detector to the network (like Virgo [?]) is required for a point sky location. BAYESTAR [?] is a tool for providing accurate sky maps from gravitational-wave events in as little as a few seconds. As a gravitational-wave signal is observed by templates with higher and higher frequency cutoffs, the sky map will become more accurate due to the increase in accumulated SNR. To demonstrate the improving sky map localisation as the signal progresses through our template bank we can produce the sky maps found by the early warning search when looking for a  $1.4 M_{\odot}$ - $1.4 M_{\odot}$  binary neutron star injection at a random point on the sky. The gravitational-wave injection is placed at a distance such that the full bandwidth SNR of the signal is 30. Figure ?? shows the sky map displaying the 90% confidence interval contours for every frequency cutoff in the template bank as well as the full bandwidth sky map, the sky location of the injection can be seen as a star on the sky map. In the legend of the figure, you can see the decreasing sky localisation area as frequency cutoff increases.

We can do the same for an injection placed at a distance such that the full bandwidth SNR is 10 and 20 and display the sky localisation areas at the different frequency cutoffs in Table ?? . The 10 SNR injection was not seen by the early warning search for frequency cutoffs 29 and 32 Hz, this is due to an event not being found

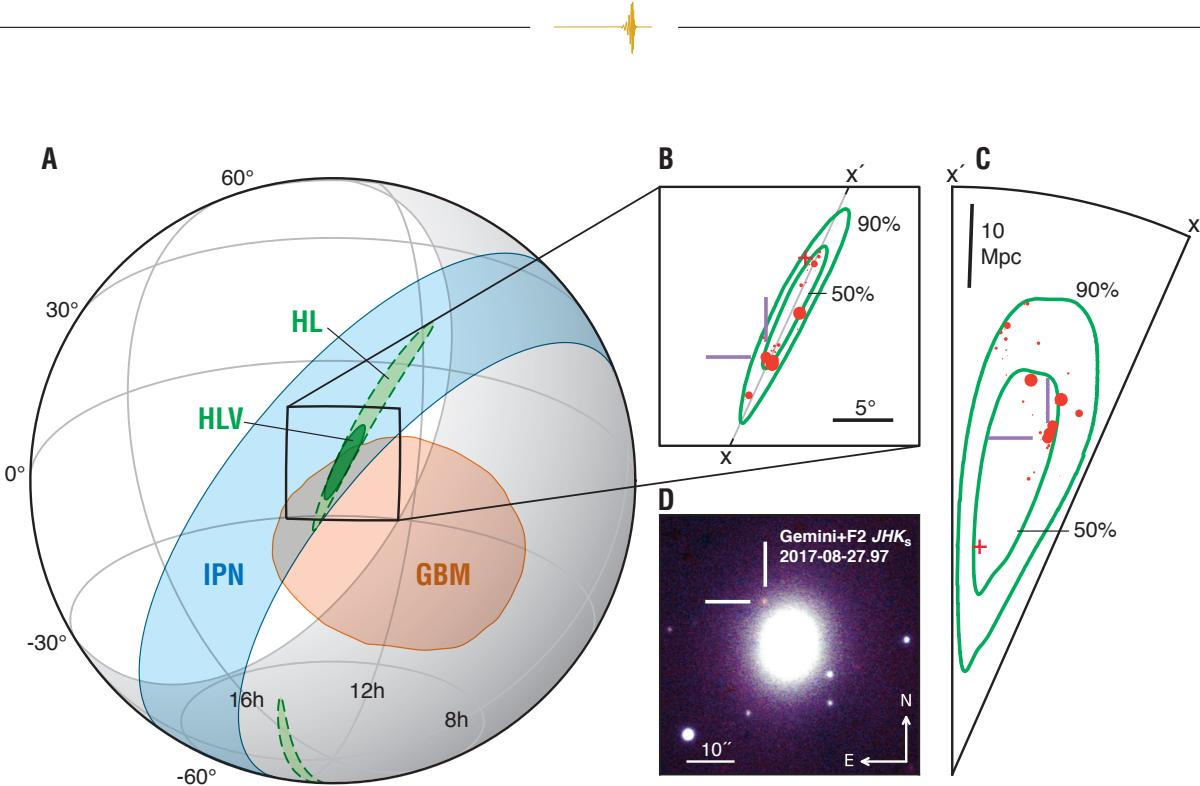


Figure 6.5: The localisation of GW170817 and associated electromagnetic counterpart. The rapid LIGO localisation is indicated by the green dashed contour, and the LIGO & Virgo localisation by solid green. Fermi [?] is shown in orange, and the Interplanetary Network triangulation from Fermi and INTEGRAL [?] in blue. This figure is taken directly from [?]

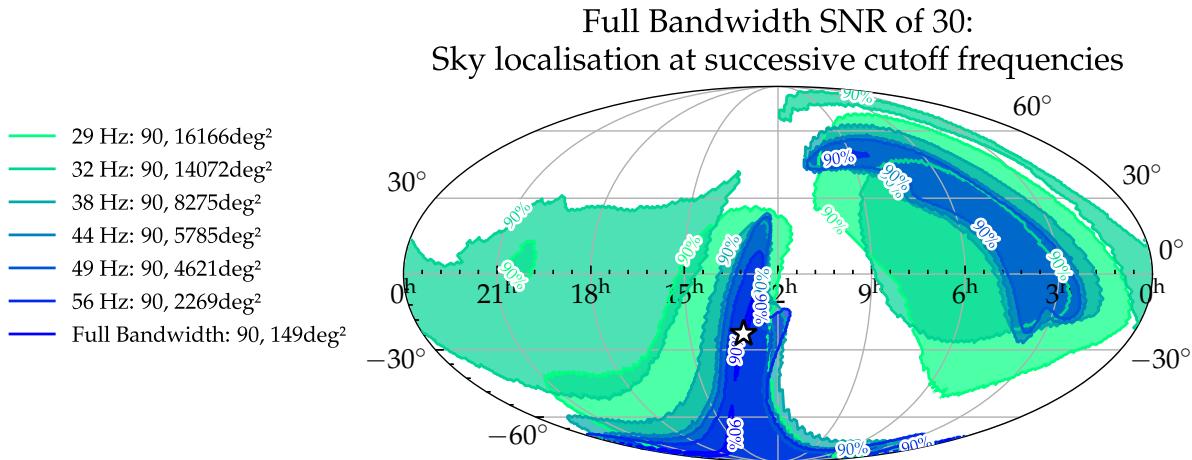


Figure 6.6: The gravitational-wave sky localisation probability map (sky map) corresponding to a  $1.4 M_{\odot}$ - $1.4 M_{\odot}$  binary neutron star signal with a full-bandwidth (upper frequency cutoff = 1024 Hz) signal-to-noise ratio of 30. The successive 90% confidence interval sky map contours for frequency cutoffs from 29–56 Hz are shown, where it can be seen that a higher frequency cutoff leads to a more accurate sky map.

---

Full SNR	10	20	30 (Fig. ??)
Frequency [Hz]	Localisation area [ $\text{deg}^2$ ] (90% credible area)		
29	Not Found	26,583	16,166
32	Not Found	22,464	14,072
38	Not Found	12,303	8,275
44	18,584	8,408	5,785
49	14,068	6,758	4,621
56	9,636	4,679	2,269
1024	1,908	300	149

Table 6.1: The sky localisation areas found by BAYESTAR [? ] for three different gravitational-wave injections injected at the same sky location but with different distances so as their full bandwidth signal-to-noise ratio to be equal to the ‘Full SNR’. The sky localisation 90% confidence interval is shown for the six frequency cutoffs found in the PyCBC Live early warning template bank [? ] as well as the sky localisation area for the full bandwidth (1024 Hz) event. Events which weren’t found above the PyCBC Live early warning search signal-to-noise ratio threshold for a frequency cutoff have no sky localisation and have been given a value of ‘Not Found’ in the table.

above the search’s SNR threshold. It is clear that a larger frequency cutoff leads to a more accurate sky localisation.

## 6.4 Detecting GW170817 in early warning

GRB 170817A was seen by Fermi 1.7 seconds after GW170817 was seen by LVK. Fermi observes the complete sky over a  $\sim$ 90 minute period [? ] and can enter a pointing mode to re-point toward high profile events, such as those reported by gravitational-wave detectors. With the GRB counterpart being the earliest confirmation of a multi-messenger event, it is even more important that gravitational-wave observatories provide as much warning for an event to allow other telescopes to be in position for detecting electromagnetic counterparts. The gravitational-wave signal from GW170817 could have been detected by a 29 Hz template 67 seconds pre-merger, that is 67 seconds warning we can provide to electromagnetic observatories.

GW170817 was seen by gravitational-wave searches and the initial GCN circular was distributed 27 minutes after merger time. The LIGO-Livingston data contained a very loud glitch which had to be removed before a sky map could be made, leading to a sky map latency of 5 hours and 14 minutes. The Q-scan containing the glitch can be seen in Figure ?? [? ]. The original event was uploaded to GraceDB (the Gravitational Wave Candidate Event Database [? ]) and can be seen at on this web



page: <https://gracedb.ligo.org/events/G298048>. In the fourth observing run we now have well tested and robust search pipelines for gravitational waves and as previously stated, we expect a maximum latency of 20 seconds post-merger [? ], glitches can be auto-gated and BAYESTAR [? ] can produce sky maps in a few seconds.

We can demonstrate the early warning search’s efficacy on GW170817. Table ?? displays the template bank cutoff frequencies alongside the network SNR of the early warning search searching for GW170817 in the original data and a GW170817-like injection in simulated data with the current detector noise curve, also shown is the time pre-merger (warning) at which GW170817 will reach the specified frequency. The single detector SNR threshold is 4.5, giving a network SNR threshold of 6.36. GW170817 was much louder in the LIGO-Livingston interferometer, LIGO-Hanford’s single detector SNR did not increase above the threshold until the 56 Hz frequency cutoff, at which the network SNR also crossed the threshold—12 seconds prior to the merger. The GW170817-like injection into the O4 noise curve has very large SNRs in all the frequencies, and the 29 Hz template was observed with a 67 second warning. By implementing an early warning search in PyCBC Live we are capable of identi-

Frequency		Network SNR		Warning
Cutoff [Hz]	Original Data	O4 Injection	Time [s]	
29	6.19	13.26	67	
32	5.76	18.58	52	
38	6.11	21.96	33	
44	6.25	25.87	22	
49	6.19	24.42	17	
56	7.04	35.24	12	

Table 6.2: The signal-to-noise ratio values for the gravitational-wave events with corresponding frequency cutoff found by the PyCBC Live early warning search when searching over the data which contains GW170817 [? ] (Original Data) and data which contains a GW170817-like injection into representative data from the fourth observing run (O4 Injection). Also shown is the time before merger in which the event was found (Warning Time).

fying gravitational-wave signals before merger. This will aid in localising potentially electromagnetically bright gravitational-wave signals and inform telescopes of all frequency ranges to capture as much of the electromagnetic counterpart as possible, enabling greater quality multi-messenger astronomy.

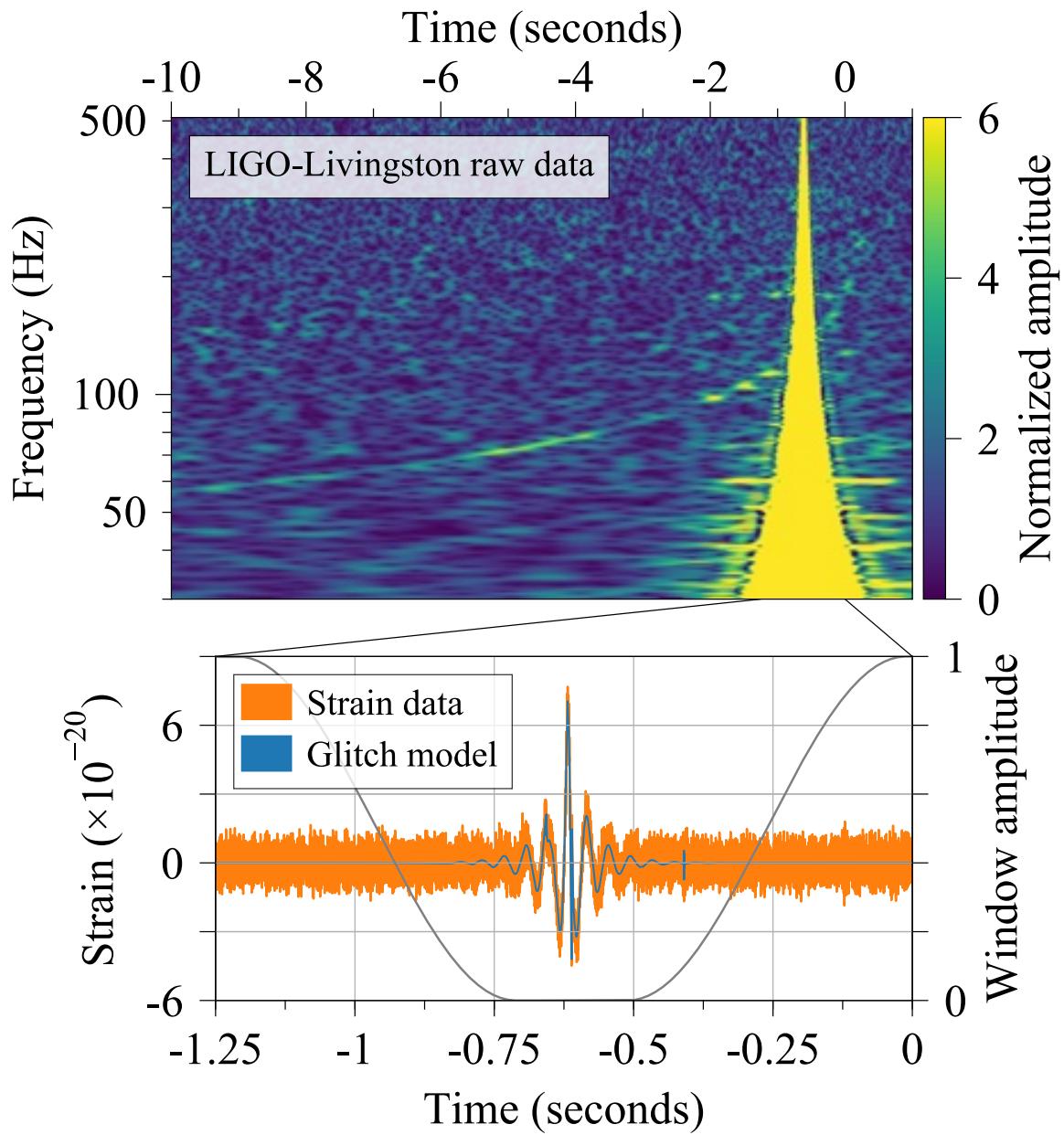


Figure 6.7: The Q-scan [?] of the LIGO-Livingston data containing GW170817 [?], with a very loud wide bandwidth glitch present (top). The glitch was initially subtracted from the data using a windowing function to reduce the amplitude of the data containing the glitch to 0, this windowing function can be seen as a grey line in the bottom figure. The glitch was also modelled using Bayeswave [?] and subtracted from the data to preserve the gravitational-wave signal power found beneath the glitch, this glitch model (blue line) can be seen overlapping the strain data (orange line) in the bottom figure. This figure has been taken from [?].



## 6.5 Testing the early warning search

The early warning search has been operating throughout the fourth observing run, and is yet to see any early warning events. A Mock Data Challenge, which cyclically re-runs 40 days of third observing run data, is being used to test the capabilities of the early warning search and has suggested that the search is not running optimally.

To thoroughly evaluate this search pipeline’s performance, we want to carry out the first large scale injection study of early warning-like gravitational-wave signals to directly test the efficacy of the search. By performing this injection study, we will identify any problems and improvements that can be made to enable the greatest possibility of detecting multi-messenger events via early warning in current and future observing runs.

In this section, we describe the injection study of the early warning search, detailing the parameters of the injection set used and the results of the study; the total number of injections found with at least one candidate event and the total number of candidate events found for all injections.

### 6.5.1 Searching for thousands of gravitational-wave injections

To test the capabilities of the early warning search an injection campaign was performed, similar to the injection campaigns performed in chapters ?? and ???. The injection parameter value ranges match the early warning template bank, therefore, the injection set used in this test contains exclusively signals that should be seen by the early warning search. The injection set contains 32,599 unique injections placed every  $\sim$ 100 seconds, encompassing 3,456,000 seconds or exactly 40 days worth of data. The signal parameters and value ranges can be seen in Table ??.

We first simulate the injections to create the injection set. The injections are then distributed uniformly in SNR between 12–60. We then inject the injections into simulated coloured noise therefore, no non-Gaussian artefacts exist in the data.

### 6.5.2 Results of the early warning injection test

We expect injections can be seen by up to six templates corresponding to the six different frequency cutoffs. With 32,599 injections in the injection set we could expect a maximum of 195,594 candidate events in the results. We can expect some deviation to this number due to false events caused by random noise and the lower frequency

---

Variable Parameters		
Parameter	Value Range	Prior Distribution
Primary Mass, $m_1$	1.0–3.0 [ $M_\odot$ ]	uniform
Secondary Mass, $m_2$	1.0–3.0 [ $M_\odot$ ]	uniform
Primary Spin z-component, $spin1z$	−0.05–0.05	uniform
Secondary Spin z-component, $spin2z$	−0.05–0.05	uniform
Phase, $\phi_c$	0– $2\pi$	uniform angle
Inclination, $\iota$	0– $\pi$	sin angle
Polarisation, $\psi$	0– $2\pi$	uniform angle
Right Ascension, $\alpha$	0– $2\pi$	uniform sky
Declination, $\delta$	$-\frac{\pi}{2}$ – $\frac{\pi}{2}$	uniform sky

Static Parameters	
Parameter	Value
Waveform Approximant	IMRPhenomXAS [?]
Lower Frequency, $f_{lower}$	17 [Hz]
Reference Frequency, $f_{ref}$	17 [Hz]

Table 6.3: The parameters used to create the injection set used to test the PyCBC Live early warning search. Variable parameters will be assigned a random value within in the value range, distributed across all injections according to the prior distribution. Static parameters are the same for all injections in the injection set. Created using PyCBC [?].

cutoff templates not finding events for low SNR injections—some injections are just too quiet to be seen.

We create and run the early warning search on the 40 days of data containing the injections for both LIGO-Hanford and LIGO-Livingston to find two-detector coincidences. This is consistent with the PyCBC Live early warning search which currently doesn’t use Virgo data to identify candidate events. The total number of candidate events found is 175,219, we can correlate a candidate event’s time of coalescence with the known injection times to eliminate 1,211 candidate events which were not found within a short 2 second window of an injection time of coalescence. The total number of candidate events found for all injections is 174,008, Figure ?? shows a histogram of the number of candidate events found per injection.

Upon inspection of Figure ?? we immediately identify an issue with the early warning search; the histogram shows us a large number of injections which have been found with greater than six candidate events. This is unexpected and we will explore the cause of this in the following sections, as well as other identified issues.

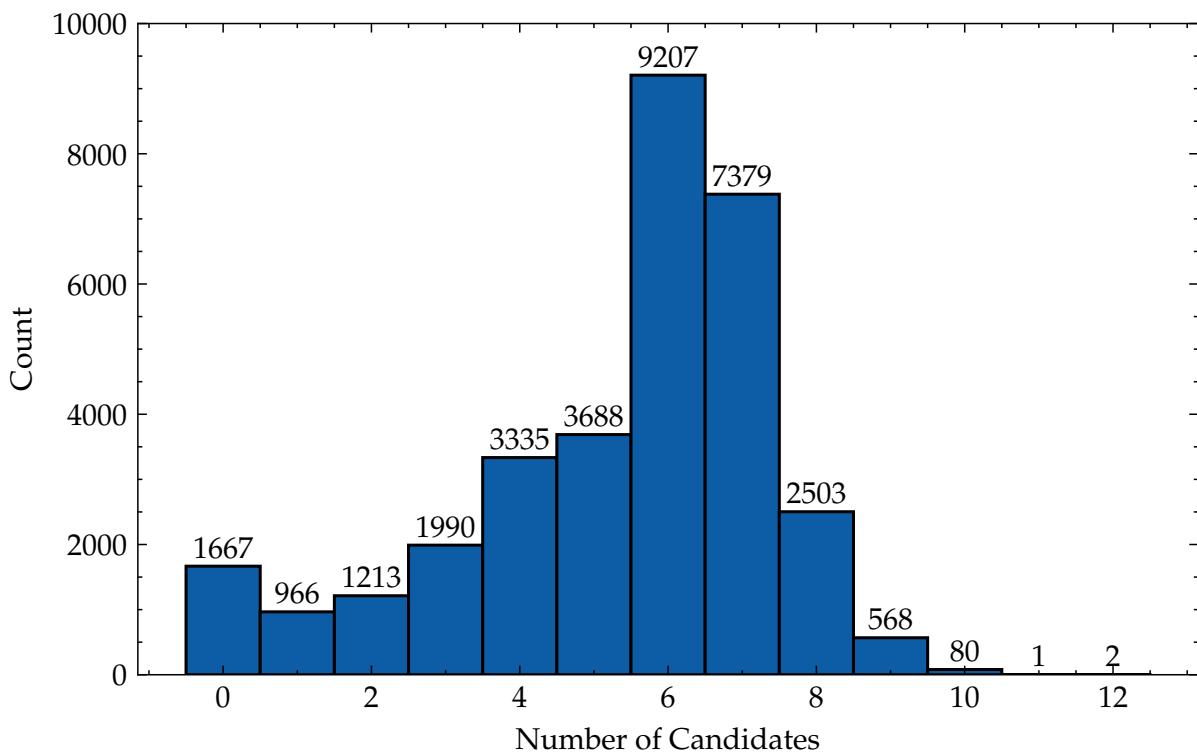


Figure 6.8: A histogram showing the number of injections (Count) found with varying numbers of candidate events (Number of Candidates) by the PyCBC Live early warning search. The exact number of counts is displayed on top of the histogram bars.

## 6.6 False events

With these results we can evaluate the performance of the early warning search and identify the problems that we initially set out to find. For each identified problem, we describe its source, estimate the number of injections affected, and detail potential solutions to be implemented into the early warning search to prevent the problem from occurring in future observing runs. We will begin with the problem identified by Figure ??, where the early warning search is identifying too many candidate events for some injections.

### 6.6.1 Finding duplicate frequency cutoff candidates

The template bank used by the early warning search contains templates for six different frequency cutoffs. The time before merger corresponding to each frequency cutoff is injection specific and it is expected that when the early warning search reaches that time, it would identify a candidate event found by a template with the correct frequency cutoff. This then places a maximum number of events found for each injection using the template bank at six events.

We have observed injections that have been seen with more than six candidate events. These candidates must have frequency cutoffs equal to one of the six found in our template bank and therefore some frequency cutoffs have been found multiple times for an injection. This should not be possible as the time before merger for the frequency cutoff would correspond to a stride in the early warning search and only the best coincident event can be chosen by the early warning search as a candidate.

#### Candidates straddling search boundaries

The early warning search operates on a single second analysis stride, meaning every single second the template bank is matched filtered with the data to discover any gravitational-wave events. It is possible that the pre-merger time corresponding to a frequency cutoff will lie very near to the boundary between search strides. In this case, the search will record an event found in the first stride and will then again record another event in the second stride with the same frequency cutoff.

Figure ?? is a demonstration of this possibility. Either event can have a higher SNR when compared to the other, but the two scenarios can be treated differently to prevent some duplicate events being uploaded in the future. If the event found in the first stride has a higher SNR than the second stride we can prevent an upload of the second event (provided the template frequency cutoff is the same)—this is already

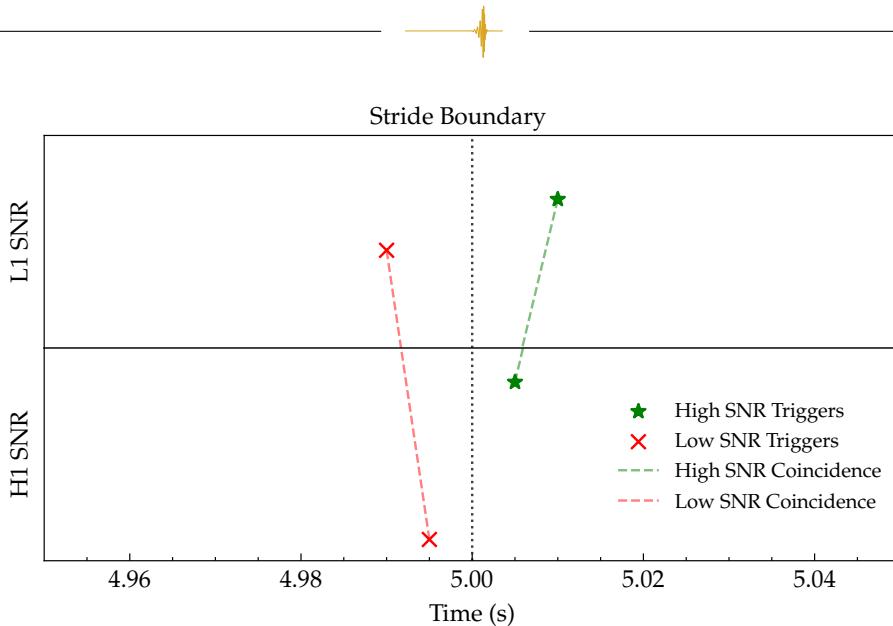


Figure 6.9: An illustration of two candidate events being found with the same frequency cutoff for a single injection. The red coincidence has lower single detector and coincidence SNR, but both are above SNR threshold due to the frequency cutoff template ending very close to the analysis stride boundary, this event is uploaded as a candidate event. The green coincidence is the ‘proper’ coincidence found after the frequency cutoff template has ended and it also uploaded after the first one, leading to two events with the same frequency cutoff being uploaded for one gravitational-wave signal.

done by the PyCBC Live Single Detector search [? ]—but if the event found in the second stride has a higher SNR than the first then we will have to upload two events. We cannot delay uploading the first event.

### Triggers across search boundaries

The stride boundary can have another effect, which causes duplicate frequency events to be uploaded. Coincident triggers between detector 1 and detector 2 have an allowed time difference in each detector to account for the physical light-travel time of the gravitational wave and a smaller amount of computational timing errors. It is possible for the two single detector triggers to appear across a stride boundary.

The trigger will first be seen in detector 1, which will be seen as a highly significant single detector trigger by the early warning search, where it will attempt to match with a coincident trigger in the other detector. As previously mentioned, this could be Gaussian noise which happens to match well enough for a low-significance coincidence to be made, or it could be the final point in a rising SNR time series. When detector 2 records the trigger in the next stride the early warning search will create the actual significant coincident event using the same detector 1 trigger twice. Therefore, we will get two candidate events at the same final frequency cutoff with one being

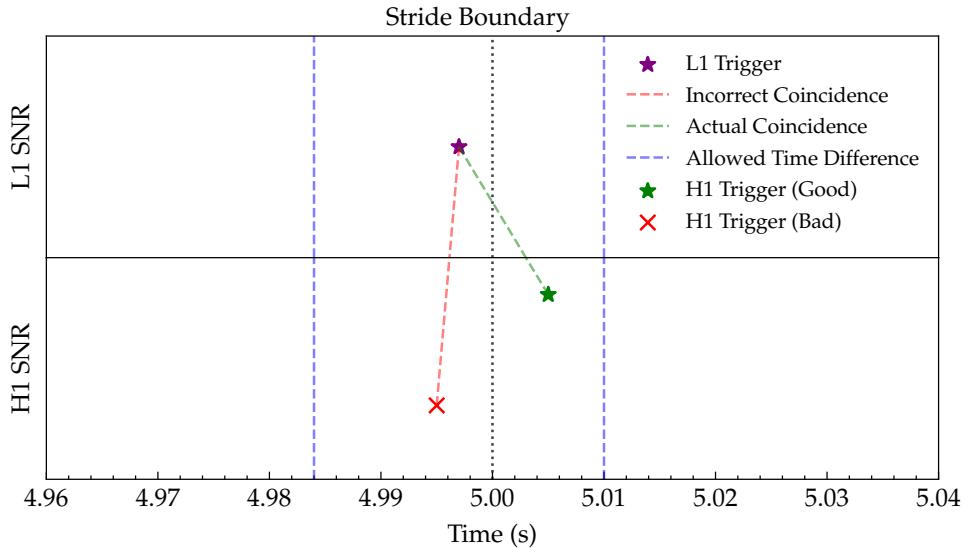


Figure 6.10: An illustration of two coincident candidate events being uploaded for the same frequency cutoff. The L1 trigger for the coincidence arrives in the analysis stride and the H1 trigger arrives in the next analysis stride. The early warning search has found a low significance H1 trigger in the first analysis stride and made a low significance coincidence. When the second analysis stride is analysed the correct coincidence will be found and the same L1 trigger is uploaded again with another coincident trigger at the same frequency cutoff.

less significant than the other. A demonstration of this case is seen in Figure ??.

This is an unavoidable failure mode of the early warning search, we require the light travel time in the ranking statistic to form coincidences between single detector triggers and this will naturally fall across the stride boundary for some injections.

We find that 13,098 injections (40.17% of the total) were found with at least one duplicate frequency and 17,432 duplicate frequencies can be removed from our results leaving 156,576 unique candidate events remaining. Figure ?? recreates Figure ?? where the duplicate candidate events have not been included.

Candidate events are all treated independently and while the additional events uploaded to GraceDB may be confusing, gravitational-wave signals on GraceDB commonly have multiple events uploaded per pipeline with multiple pipelines. Therefore, the event with the highest SNR will be preferred until a higher SNR event is identified.

## 6.6.2 Low significance candidate events

We have discussed false alarm candidate events that do not have times of coalescence consistent with any known injection time of coalescence and have removed 1,211 candidates from our final results. There is a possibility of a false alarm candidate event

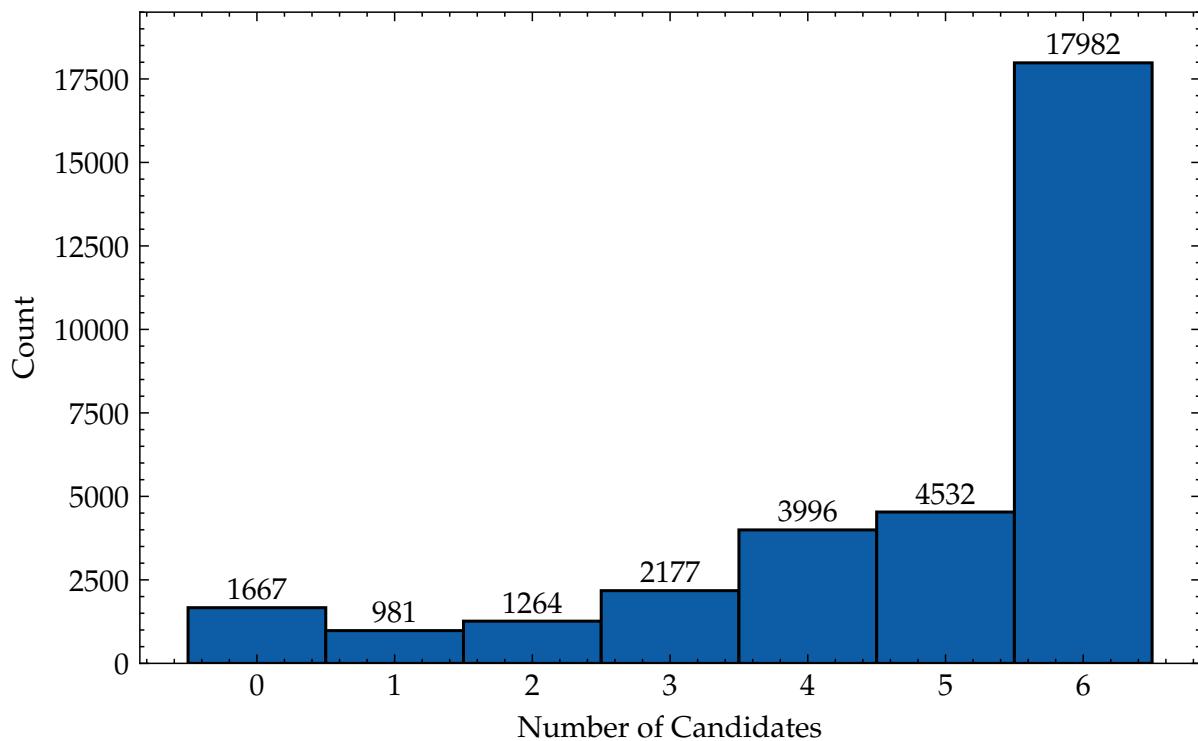


Figure 6.11: A histogram showing the number of injections (Count) found with varying numbers of candidate events (Number of Candidates) by the PyCBC Live early warning search. The exact number of counts is displayed on top of the histogram bars.

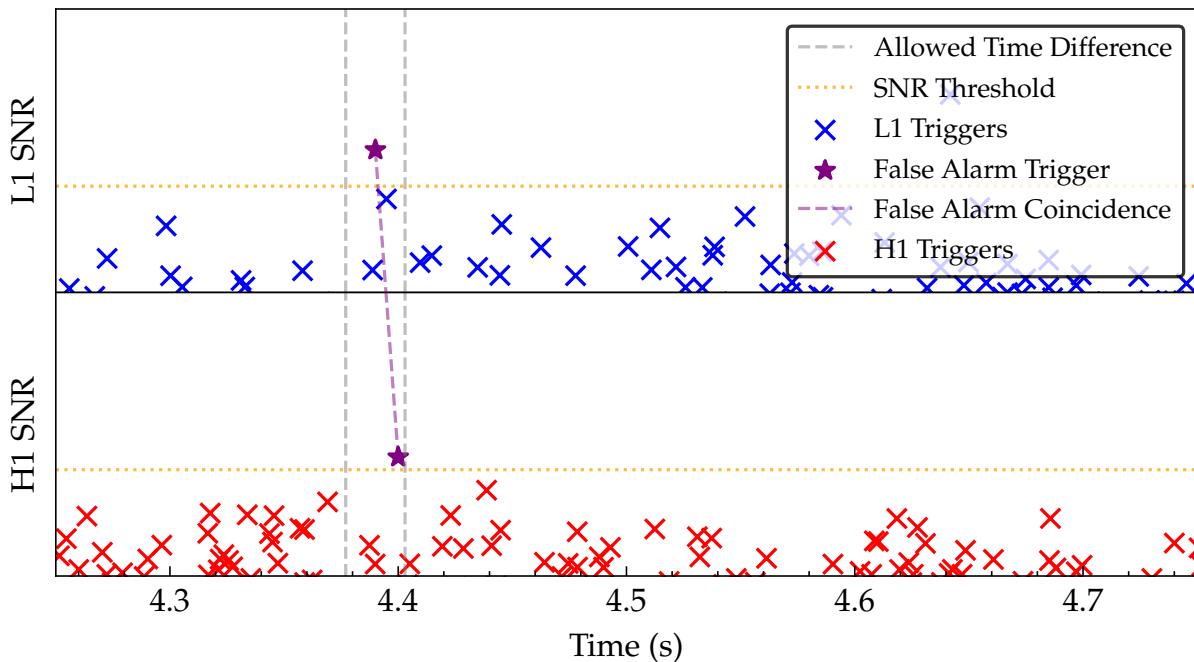


Figure 6.12: An illustration of a coincident candidate being found by the alignment of Gaussian noise triggers in two detectors. The crosses represent the single detector triggers identified by the gravitational-wave search, triggers found above the SNR threshold can be considered a single detector trigger. Coincident triggers have to lie inside the allowed coincidence window, shown by the grey dashed lines around the L1 trigger.

to have a time of coalescence that falls nearby to an injection. These are easy to identify because those that appear at a lower frequency will not have the corresponding candidates from the next frequencies in the template bank. This is not something the early warning search can take into account in low-latency, but it is something that can be used post-detection to confirm the legitimacy of an event. The injection set used in this search was made using a PSD representative of the advanced LIGO noise curve [?] and therefore is coloured Gaussian noise, not containing any non-Gaussian noise transients. This means that when we identify a false-alarm candidate it has been caused by Gaussian noise triggers aligning in between the two detectors and finding a coincidence, a demonstration of this can be seen in Figure ??.

We can identify further false-alarm candidate events in two ways: look for injections with single candidates that have frequency cutoffs not equal to the highest cutoff in the template bank (56 Hz); and by comparing the injection template and the candidate event template. We observed 99 injections with single candidates and a further 39 candidate events with a mismatch between template  $\mathcal{M}$  and injection  $\mathcal{M}$  of  $\pm 1\%$ .

Of the original 174,088 candidate events identified and we have been able to eliminate 17,570 candidate events that were caused by boundary issues or false-alarms,

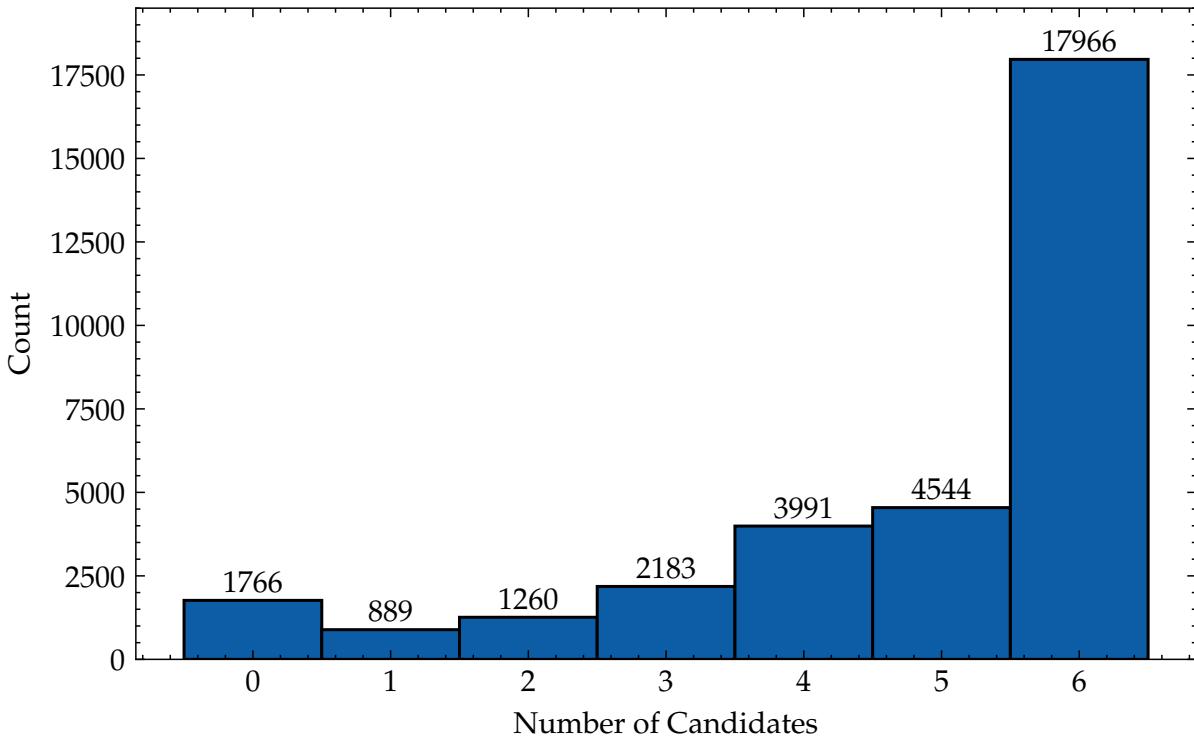


Figure 6.13: A histogram showing the number of injections (Count) found with varying numbers of candidate events (Number of Candidates) by the PyCBC Live early warning search. The exact number of counts is displayed on top of the histogram bars.

leaving 156,438 candidate events identified by the early warning search during the injection study. Figure ?? shows the distribution of the number of candidate events found per injection, where 13,225 injections have additional unexpected candidate events.

## 6.7 Candidate events we did not see

We have identified injections that have seen extra candidate events from what we expect and now we can look at injections where we expected to see more events than we did. We count the number of candidate events for each injection and compare to the expected number of candidate events considering the lowest frequency cutoff the injection was found with. For example, if an injection has been seen at 29 Hz then we would expect six candidate events in total, whereas for an injection seen first at 49 Hz we would only expect two candidate events (49 Hz and 56 Hz). A further inspection of these candidate event SNRs can reveal another issue with the observations, we expect a monotonic increase in the SNR of each subsequent frequency cutoff in the template

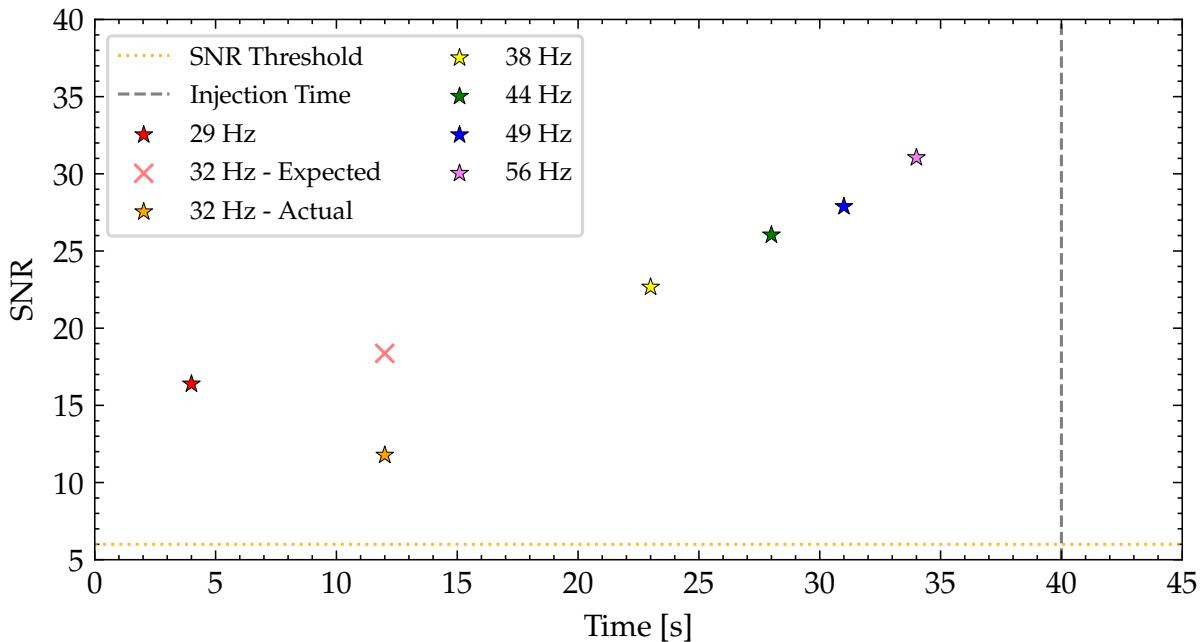


Figure 6.14: An illustration of the expected and SNRs for the case where one of the event’s coincident triggers have fallen outside the coincident timing window. In the first case we see a missing frequency in the SNR evolution (indicated by the red cross), in the second case we see a non-monotonic increase in SNR where the search has found a low significance event to ‘fill’ the gap (shown by the orange star).

bank. If the SNR between consecutive candidates does not exhibit this SNR increase then we know this injection has encountered a problem we must investigate. Figure ?? shows an illustration of these injections, where the 32 Hz has either not been seen at all or has been seen with a much lower SNR than what is expected.

### 6.7.1 Coincident triggers outside the light travel time

We investigate the frequency of the missing candidate events for injections that have a gap in frequency evolution and obtain the following counts per frequency cutoff: 29 Hz = 0 (We do not distinguish between missed and not found for the first frequency cutoff), 32 Hz = 288, 38 Hz = 162, 44 Hz = 132, 49 Hz = 168, 56 Hz = 140. Some injections have multiple missing frequencies, and in total we find 873 injections (2.68%) are missing at least one expected candidate event in its frequency evolution. This is a lower limit on the true number of injections with missing candidate events, we are unable to count injection where: the 56 Hz template was the only one above the SNR threshold and was missed; the missed frequency is the lowest frequency in the evolution, for example 29 Hz, then we are just assuming that the 29 Hz did not have enough SNR to be seen and wasn’t ‘missed’ in this context.



The early warning search will output all the triggers found when matched filtering the template bank and the data every second. Candidate events are detected in the analysis segment containing the end time of the template, which represents the frequency cutoff for the injection. Therefore, we can look at the triggers found by the search to determine why a candidate event was not identified. The injections with missing frequencies all record single detector triggers above the SNR threshold and with the best matching template for both detectors but, we have identified that the time difference between these two triggers is greater than our allowed coincidence timing window—a component of the coincident ranking statistic to demand physical time differences between triggers.

The two LIGO detectors are separated by a straight line through the Earth 3002 kilometres long. Given the speed of light there is a 0.01 second window for the gravitational-wave signal to travel from one detector to the other. We allow a further 0.003 seconds in addition to the light travel time to account for any detection response, timing, or waveform uncertainties. This give a total allowed coincidence timing window of 0.013 seconds. An illustration of this problem can be seen in Figure ?? . How is it possible for our ‘coincident’ triggers from our injection set to lie outside the allowed coincidence timing window?

### Timing accuracy error

As described in [? ], the accuracy of the candidate time in a detector can be determined by,

$$\sigma_t = \frac{1}{2\pi\rho\sigma_f} \quad (6.12)$$

where the timing accuracy,  $\sigma_t$ , is inversely proportional to the SNR,  $\rho$ , and the effective bandwidth of the source,  $\sigma_f$  which is defined as

$$\sigma_f^2 = \left( \int_0^\infty df \frac{f^2 |\hat{h}(f)|^2}{S(f)} \right) - \left( \int_0^\infty df \frac{f |\hat{h}(f)|^2}{S(f)} \right)^2. \quad (6.13)$$

where the waveform  $\hat{h}$  is normalised such that  $\int \frac{|\hat{h}(f)|^2}{S(f)} df = 1$ . We use the same PSD to normalise that has been used throughout the injection study. As shown in Table 1. of [? ], the timing error for a binary neutron star system in an advanced LIGO configuration with a BNS range of 160 Mpc is 0.46 ms. This is negligible when compared to our light-travel time of 10 ms. We can approximately calculate the timing accuracy of the early warning templates using equations ?? and ??, where the gravitational-wave strain term,  $|h(f)|$ , is assumed to be equal to the inspiral frequency evolution

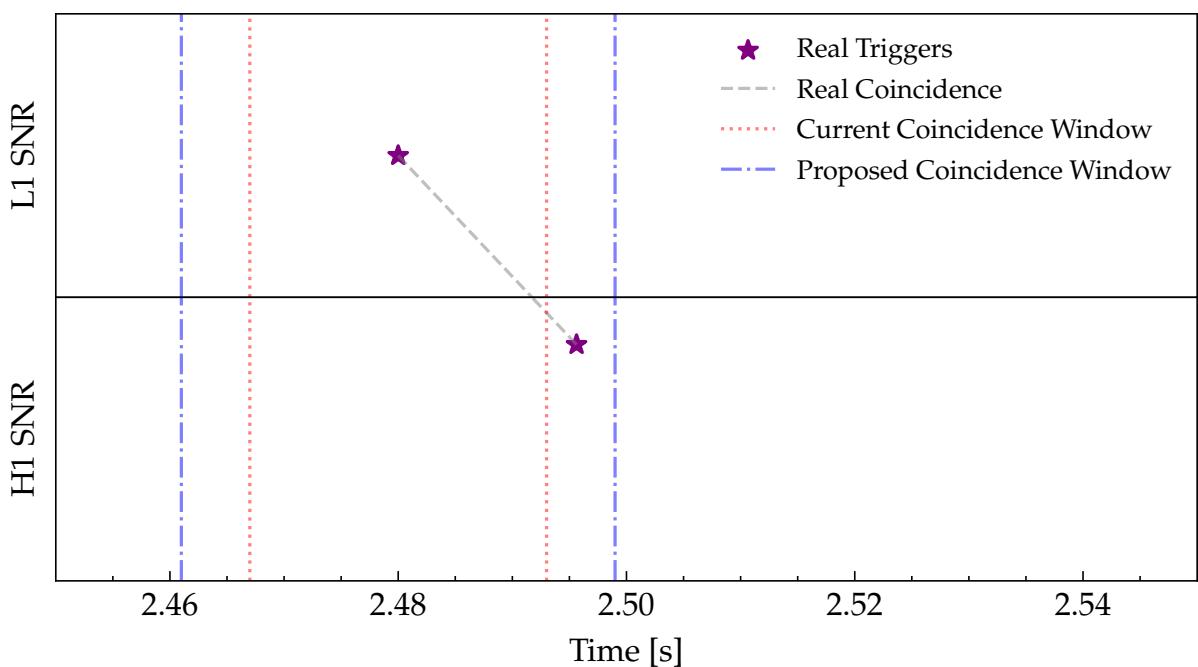


Figure 6.15: An illustration of two single detector triggers being found outside the allowed coincidence time difference window. The current coincidence window shows the triggers falling outside the window and therefore the search does not form a coincidence. With the new proposed coincidence time window the coincidence would be found.



of a binary neutron star signal,  $f^{-\frac{7}{6}}$ . The estimated effective bandwidths and timing accuracy errors can be seen in Table ???. Using an SNR of 8 (and a lower frequency

Frequency [Hz]	$\sigma_f$ [Hz]	$\sigma_t$ [ms]
29	3.38	5.89
32	4.19	4.75
38	5.79	3.44
44	7.37	2.70
49	8.68	2.29
56	10.51	1.89
1024	90.05	0.22

Table 6.4: The effective bandwidth,  $\sigma_f$ , and timing accuracy error,  $\sigma_t$ , for the frequency cutoffs (Frequency) found in the early warning template bank as well as the full bandwidth frequency (1024 Hz).

of 17 Hz) yields 1 sigma timing errors from 5.89 to 1.89 milliseconds for the early warning frequency cutoffs, where higher frequencies have smaller timing errors. The current allowed error is 3 ms therefore we can expect to miss candidates from all the frequency cutoff templates when the timing error places coincident triggers outside the current allowed coincidence time window. This can be remedied with a widening of the window which could be done on a template dependent basis, with a timing accuracy error dependent on the frequency cutoff.

The PyCBC Live early warning search uses a sample rate of 256 Hz which contributes to a worse timing accuracy. The time between samples is 3.9 ms and a trigger found by another detector 3 samples later will be within the allowed window of 13 ms but being 4 samples later equals a time difference of 15.6 ms; falling outside the window. We can perform a subsample interpolation between samples to recover the correct trigger time within the allowed window.

### Early warning tuned Phase-Time-Amplitude histogram

Increasing the allowed coincidence time window allows the early warning search to form coincident triggers which share the same template parameters and are physically possible (when accounting for timing accuracy error). However, we will face another problem where the coincident ranking statistic used by the early warning search will consider these events to have a very low signal rate. The coincident ranking statistic used by the early warning search is phasetd (phase-time delay) and is used to assess the coincidence likelihood between two triggers from different detectors.

For a two detector coincidence, the likelihood value is found by sampling from a pre-created phasetd 3-dimensional histogram in amplitude, time, and phase space.

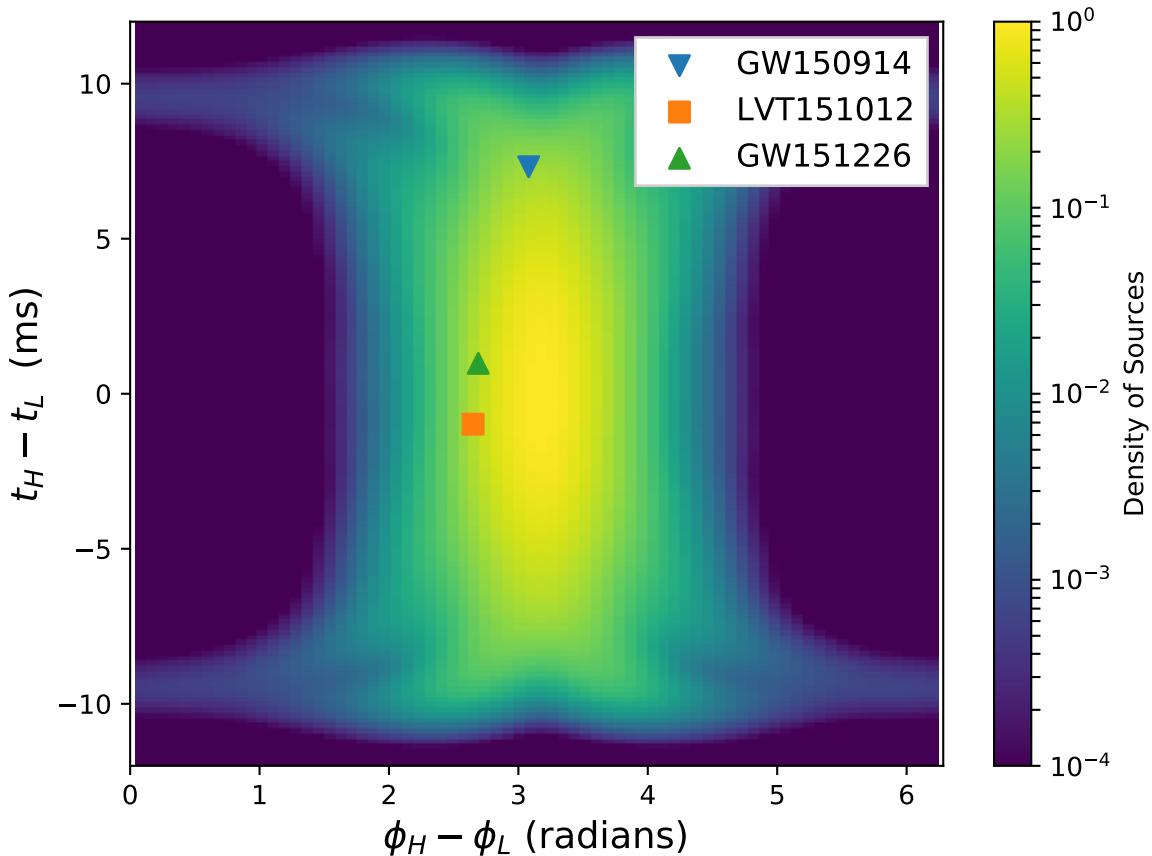


Figure 6.16: Phase-Time-Amplitude histogram for the PyCBC ranking statistic, showing the density of gravitational-wave candidate sources as a function of time delay on the y-axis,  $t_H - t_L$  in milliseconds, and phase difference on the x-axis,  $\phi_H - \phi_L$  in radians, between detectors. The first three gravitational-wave candidates have been highlighted on the plot [? ]. The colour scale indicates the density of sources in logarithmic scale. The histogram can be used to determine how likely a signal is based on the differences in phase, time, and amplitude recorded by each detector. Taken from [? ].



The file is created by simulating gravitational-wave signals from an isotropic distribution of sources, where each signal is assigned a random sky location (right ascension and declination), inclination and polarisation. The detector responses to the signal are calculated, from which we get the amplitudes, time, and phases for each injection in each detector. For a pair of detectors, the time difference, phase difference and relative signal amplitude are then binned into discrete bins which represent different combinations of amplitude, time, and phase difference. An example of the phase-time histogram (marginalised over amplitude) can be seen in Figure ?? [? ] where the density of sources at time differences greater or less than 13 ms is extremely low. Therefore, when we sample from the histogram with large time difference, we will receive very low value of the signal rate. The timing accuracy error will need to be taken account explicitly in the time difference on both trigger times.

Another improvement that can be made in the generation of the `phasetd` statistic files is including the sample rate of the search as a parameter. For a Monte Carlo simulation, the trigger time differences are as accurate as a `float64` parameter can be. Whereas the timing difference between triggers in the PyCBC Live search will only take discrete values of the time difference dependent on the sample rate, for example, the early warning search has a sample rate of 256 Hz and therefore the time difference between samples is approximately 3.9 milliseconds. Therefore, our `phasetd` file will only ever been sampled from using time difference that are  $n$ -multiples of this time between samples. We can create `phasetd` files with bins in the time difference parameter space which represent the sample rate of the search and therefore can be sampled more accurately.

## Missing candidates

We know now the search will produce triggers that fall outside the allowed coincidence timing window and fail to form a coincident event. Figure ?? shows an example of an injection which has been found with all but 1 final frequency cutoff template. To verify the time difference problem, we investigated the expected network SNRs for this injection at each final frequency cutoff in both a noiseless and expected noise regime (not performed using the PyCBC Live search). The values for these can be seen in Table ?? . We can clearly see that we are expecting to see a large network SNR for the 32 Hz template, which isn't recovered by the search. Next we are able to re-run the early warning search using a template which perfectly describes the injection parameters and in this case we still fail to recover the candidate event but, we can look into the trigger files to identify the single trigger found by both searches with the same template above the SNR threshold. We find a H1 trigger with SNR 14.55

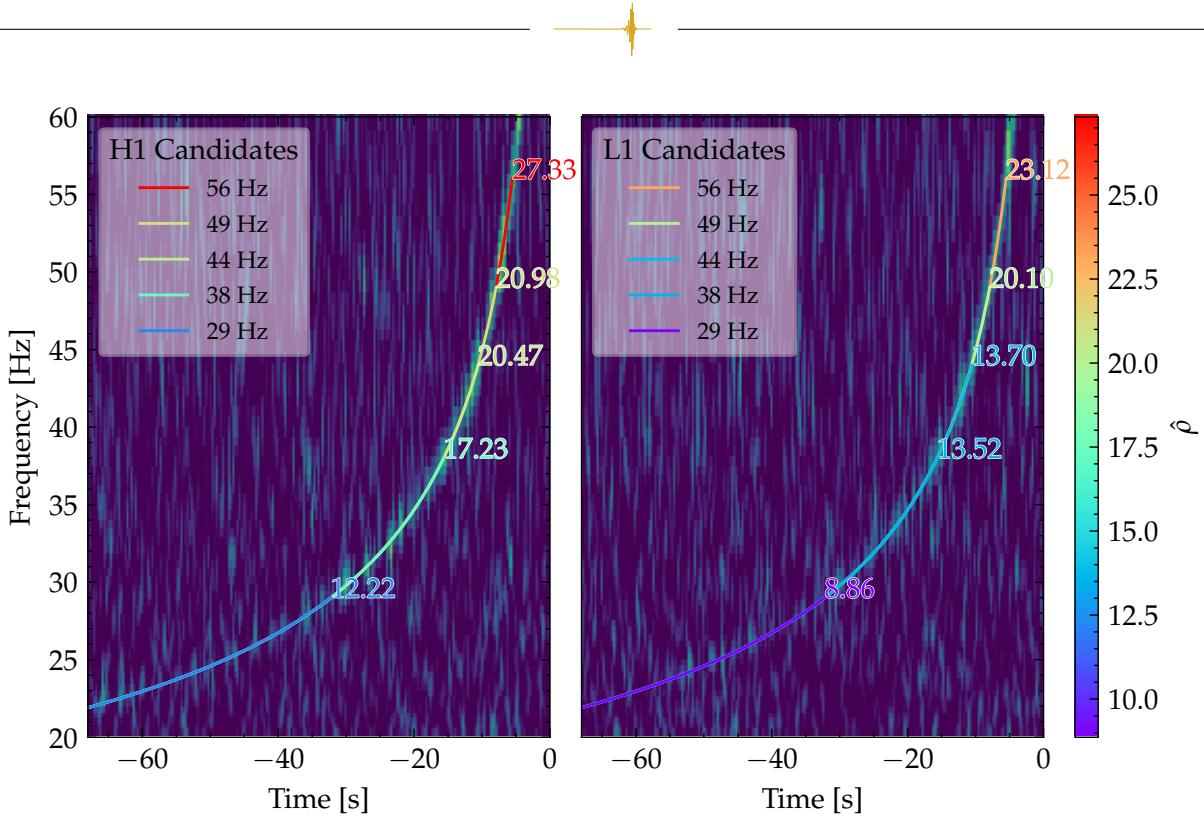


Figure 6.17: An example of a gravitational-wave injection seen by the PyCBC Live early warning search with a missing template with frequency cutoff 32 Hz.

and an L1 trigger with SNR 12.38 (giving a network SNR of 19.11) and when the time difference between triggers is calculated we get 15.625 ms, greater than the currently allowed 13 ms.

We then perform another search with a sample rate of 2048 Hz, which doesn't find a coincidence but has a smaller time difference of 14.6 ms (still outside the window) proving that the time accuracy error is playing a part and it isn't simply a sampling inaccuracy. When performing a final search in which we expand the allowed coincidence timing window by an additional 10 ms (this can be tuned in the future) we successfully recover all 6 events for this injection.

### Non-monotonic SNR evolution

Another manifestation of the timing accuracy error is the possibility for the missing frequency to be found but with a SNR not following a monotonic SNR increase across the entire injection event timeline. A demonstration of this can be seen in Figure ?? and an example of this happening for an injection in our early warning search can be seen in Figure ??.

It can be seen that the 32 Hz has been found by the search for this injection, but with a much lower SNR than expected and not following the monotonic SNR increase

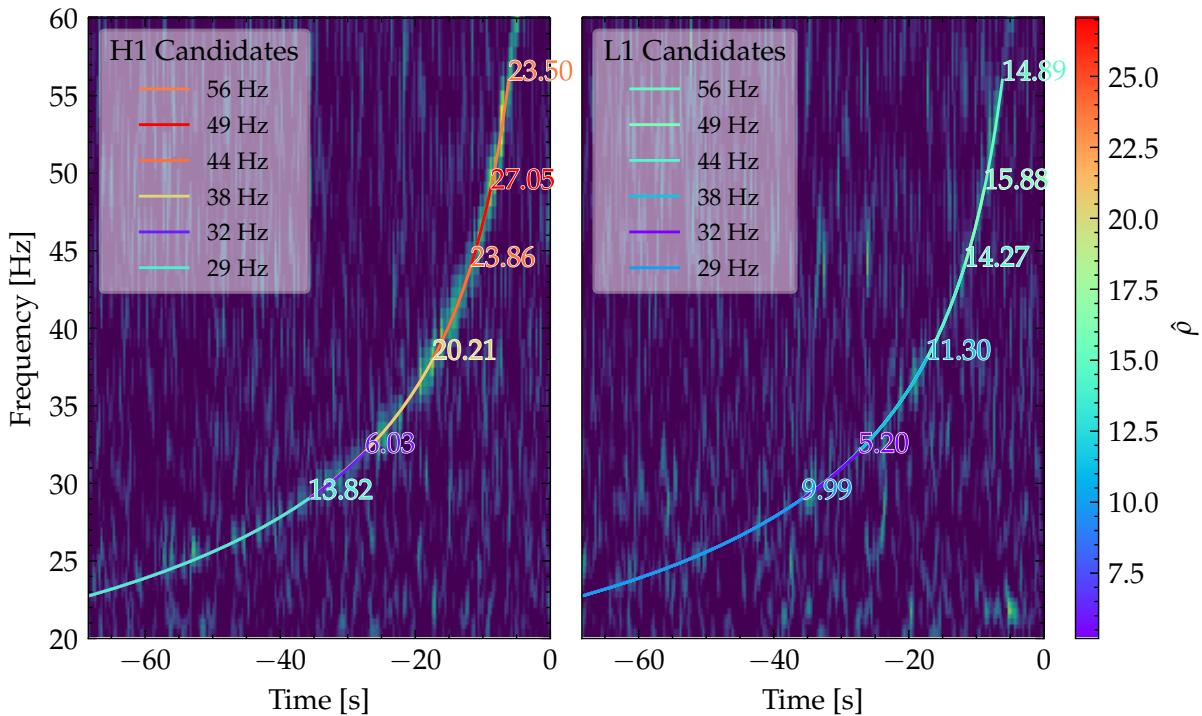


Figure 6.18: An example of a gravitational-wave injection that has been found with a non-monotonically increasing signal-to-noise ratio between successive frequency cutoffs. The 32 Hz frequency cutoff gravitational-wave template has been seen with a lower SNR in the H1 and L1 single detector searches than the 29 Hz frequency cutoff template. This is caused by the actual 32 Hz template coincident triggers being found outside the allowed coincidence timing window.

---

Frequency Cutoff [Hz]	Noiseless SNR	Noisy SNR
29	16.43	15.38
32	19.16	17.95
38	24.00	21.50
44	28.11	25.26
49	31.04	27.94
56	34.48	29.02

---

Table 6.5: The expected signal-to-noise ratios in both noiseless and noisy data for the gravitational-wave injection shown in Figure ???. The noiseless SNR has been calculated by taking the un-normalised matched-filter of the template with itself [? ]. The noisy SNR has been calculated by matched-filtering the template with the injection that has been injected into simulated gravitational-wave data.

that we would expect. When investigating the triggers found we can see that the search has managed to find a low-significance trigger within the coincidence time window and has managed to form a coincidence with low SNR, similar to the low significance candidates described in Section ?? but where one trigger is actually real. Looking into the single detector trigger files we can once again find the coincident trigger pair that should've been made if the coincidence timing window was larger and when running the search over this injection with the larger coincidence time window we find all 6 events for this injection with the expected SNR.

We find that 997 injections had non-monotonically increasing SNR between all successive frequency cutoffs. However, we expect a number of injections to have non-monotonically increasing SNRs due to Gaussian noise in the detector influencing the SNR values recovered for some templates. Therefore, we only count injections in which the drop between successive SNRs is greater than 5% of the previous frequency cutoff template's SNR. When applying this limit, we obtain only 120 injections with non-monotonically increasing SNR (0.37%).

## 6.8 Conclusion

We injected 32,599 simulated gravitational-wave signals into 40 days of simulated gravitational-wave data and searched through this data using the PyCBC Live early warning search. The PyCBC Live early warning search found 175,219 candidate events, 174,008 of these candidate events can be associated with an injection in our injection set and after removing extra unexpected candidate events from duplicate frequency events and false-alarms we obtain 156,438 candidate events for all injections.

The more significant problem identified by this injection study is the missed events



that we should've seen but were didn't due to the inadequate allowed coincidence timing window between coincident triggers. We identify 873 injections that were missing at least one expected candidate event and 120 more which missed the coincident event but found a far less significant coincidence with a significant deviation from the expected monotonic SNR increase. In total, we missed 1,766 injections completely however, of these we have been unable to estimate those missed due to the coincidence timing window.

To prevent gravitational-wave signals from being missed by the PyCBC Live early warning search a configuration change is needed to increase this window alongside an improved phase-time-amplitude histogram which accounts for coincident gravitational-wave triggers at these extended time differences.

In conclusion, the early warning search is more than capable of observing gravitational-wave signals tens of seconds prior to the merger of the gravitational-wave event. We are able to disseminate information about this event to the international scientific community however, a statistic change is required in the current and future observing runs to enable the identification of gravitational-wave signals with time differences between detectors outside the current allowed coincidence timing window. While the early warning search is more than capable of observing these events, electromagnetically bright events have proven to be exceedingly rare and therefore we do not expect to observe an early warning event in the near future. Additionally, the early warning search is robust in its capability to detect up to five individual candidate events per gravitational-wave event therefore, if we miss a single one of these due to timing differences then there is a still a high likelihood of observing the gravitational-wave event in early warning via the other four frequency cutoffs.

# 7

---

## Optimising Signal-to-Noise Ratio in PyCBC Live

---

*“Are we there yet?”*

— Donkey, Shrek 2

The work that forms this chapter was performed as my first initial project upon starting my PhD and small additions were made over the course of the PhD. The PyCBC Live SNR optimiser has not been detailed in any PyCBC methods publication to date and will form a section in the upcoming fourth observing run PyCBC Live methods paper where these changes will be published. I was responsible for tuning the current PyCBC Live SNR optimiser, developing and introducing a new optimiser to PyCBC Live, and implementing astrophysically-motivated bounds on the spin template parameters. The PyCBC Live search pipeline identifies gravitational-wave signals in real-time from the gravitational-wave data provided by the current international gravitational-wave network. Sky maps are produced of the found gravitational-wave events, which indicate the sky position of the source of the signal. The sky localisation is more accurate for an event with a greater signal-to-noise ratio (SNR), therefore, as part of the pipeline we rapidly optimise event SNR using an optimisation algorithm to recover the greatest SNR and more accurate sky localisation possible.

There are many optimisation algorithms with many tuneable hyperparameters which determine operating parameters of the algorithm, the appropriate algorithms



and hyperparameter values need to be chosen to ensure the optimisation performs optimally. The optimal methods for optimising SNR in PyCBC Live were researched and improved multiple times across the course of the PhD. These changes are being written up in the PyCBC Live fourth observing run methods paper.

## 7.1 Introduction

As described in chapters ?? and ?? the PyCBC Live search pipeline (among others) is responsible for detecting and producing accurate sky maps of gravitational-wave signals in low-latency. Search pipeline events are uploaded to the Gravitational-Wave Candidate Event Database [? ] (GraceDB) from which a *preferred event* is chosen to send a GCN Circular [? ] (a rapid astronomical bulletin submitted by and distributed to astronomy community members worldwide). The preferred event is chosen based on a listed selection criterion. These rules are taken verbatim from [? ]:

When multiple online searches report events at the same time, the preferred event is decided by applying the following rules, in order:

- A publishable event, meeting the public alert threshold, is given preference over one that does not meet the threshold.
- An event from CBC searches is preferred over an event from unmodeled Burst searches (see Searches for details on search pipelines).
- In the case of multiple CBC events, three-interferometer events are preferred over two-interferometer events, and two-interferometer events are preferred over single-interferometer events.
- In the case of multiple CBC events with the same number of participating interferometers, the event with the highest SNR is preferred. The SNR is used to select the preferred event among CBC candidates because higher SNR implies better sky location and parameter estimates from low-latency searches. In the case of multiple burst events, the event with the lowest FAR is preferred.

The PyCBC Live search is a CBC search and when analysing the same number of interferometers as other CBC searches there is a tie-breaker for preferred event dependent on the SNR of the event.

It is clear from these set of rules that to become the preferred event with the most accurate sky map, we must upload the event with the highest SNR. This is where the PyCBC Live SNR optimiser becomes relevant. We construct our template bank with a minimum match of 0.97, allowing up to a 3% SNR loss for any event. If we want

---



to maximise the SNR of our event, we must perform a finer search over the template bank parameter space around the initially found template to recover that potentially lost 3%.

The SNR optimiser uses optimisation algorithms which tune input parameters (template parameters) to maximise (or minimise) an output (network SNR). There are a number of different optimisation algorithms operating under different mathematical optimisation techniques, all of which have hyperparameters which can be tuned to the problem space to optimise optimiser performance—both output value and computational time.

We detail the investigation into an optimal optimisation algorithm for PyCBC Live along with the tuning of the hyperparameters for this optimisation algorithm, how these algorithms were tested and the results of using the SNR optimiser on a number of PyCBC Live events.

In Section ?? we discuss how the SNR optimiser fits into the PyCBC Live search pipeline and when it is run, alongside this we detail the generic form of the optimisation algorithms and the current optimiser being used in PyCBC Live. In Section ?? we list the parameter value bounds for the optimised parameters and how these were calculated, in Section ?? we detail the *differential evolution* optimisation algorithm and in Section ?? how the hyperparameter of this algorithm were tuned and the improvements from tuning the hyperparameter. In Section ?? we explore other optimisers and describe Particle Swarm Optimisation, the newly implemented optimisation algorithm. We detail some additional improvements to the SNR optimiser in Section ?? and in Section ?? we analyse the average SNR increase of the SNR optimiser over the events uploaded by PyCBC in 2024. We conclude in Section ?? and detail potential future improvements to the SNR optimiser.

## 7.2 Optimising SNR in low-latency

The PyCBC Live search will matched filter the entire template bank with the data ring buffer every analysis stride (see chapter ?? for greater detail). The matched filter will produce SNR triggers when any template has been found with a matched filter above an SNR of 4.5. The PyCBC Live search will then pass these triggers to the ranking statistic, taking into account any signal-consistency test values or other ranking statistic components, to find the most significant coincident set of triggers. After the uploading of an event to GraceDB, the PyCBC Live search will spawn a subprocess and the PyCBC programme `pycbc_optimize_snr` is initialised. This programme takes the gravitational-wave data and is given the event template parameters to run a finer



search over the parameter space, specifically optimising for the network SNR. This finer search is performed not using a template bank approach but with an optimisation function, a process that finds the ‘best’ possible solution by maximising for a particular objective given input parameters that can be tuned. For PyCBC Live the objective being maximised is the network SNR and the tuneable input parameters are the template bank parameters: primary component mass  $m_1$ , secondary component mass  $m_2$ , primary spin z-component  $s_{1z}$  and, secondary spin z-component  $s_{2z}$ . Boundaries are applied to the input parameters, which are calculated using the initial event parameter values.

We run this SNR optimisation to produce better sky maps for the GraceDB ‘super event’ (the gravitational-wave signal each pipeline event has observed). An example for a recent gravitational-wave signal (S240924a [? ]) can be seen in Figure ???. When all pipelines pass the public alert threshold and use the same number of interferometers, the tie for preferred event is broken by the SNR of the event. For S240924a, we can see the PyCBC event (which was the preferred event) was an SNR optimised event, indicated by the label ‘SNR\_OPTIMIZED’ in Figure ??.

The original PyCBC SNR optimiser was introduced during the second half of the third observing run to improve the sky maps that are generated by PyCBC Live using the SciPy [? ] optimiser differential evolution [? ]. Examples noted in the original GitHub pull request [? ] see increases in network SNR from 9.818 to 10.149, an increase of 3.6%.

Optimisation algorithms can generally be categorised into two main types based on their approach to searching the parameter space: population-based algorithms and single-solution algorithms. Population-based algorithms initialise a set of candidate solutions (a population) and explore the parameters space iteratively using interactions between the candidates to locate the optimal solution. Single-solution algorithms use a single candidate point that is iteratively improved based on some criteria. Only population-based algorithms have been considered for the PyCBC Live SNR optimiser for their computational efficiency in parallelised computing environments where the network SNR of each point can be computed in parallel using a large number of computing cores. These population-based algorithms all follow the same general steps in optimisation.

1. Initialise a population of size  $N$  with random parameter values
2. Evaluate the network SNR of the point
3. Adjust each point in the population using the base optimisation algorithm
4. Re-evaluate the network SNR of each point

The screenshot shows the GraceDB public webpage for superevent S240924a. At the top, there is a navigation bar with links for GraceDB, Public Alerts, Latest, Search, Documentation, and Login. A message "Please log in to view full database contents." is displayed. Below the navigation bar, the superevent ID "S240924a" is shown, along with links for Log Messages and Full Event Log. The main content area is divided into two sections: "Superevent Information" and "Event Information".

Superevent Information	
Superevent ID	S240924a
Category	Production
FAR (Hz)	2.462e-09
FAR ( $\text{yr}^{-1}$ )	1 per 12.869 years
$t_0$	1411171414.55
$t_{\text{end}}$	1411171415.55
Submitted ▾	2024-09-24 00:03:30 UTC
Links	<a href="#">Data</a>

Event Information	
Group	CBC
Pipeline	pycbc
Search	AllSky
Instruments	H1,L1,V1
Event Time ▾	1411171414.546
FAR (Hz)	2.462e-09
Submitted ▾	2024-09-24 00:04:10 UTC

Figure 7.1: The public webpage for the superevent gravitational-wave signal observed on the 24<sup>th</sup> September 2024, S240924a [? ]. The preferred search pipeline for this event can be seen in the table titled ‘Event Information’, where it can be seen that a PyCBC Live is the preferred event.



GW Event Upload Information									
UID	Group	Pipeline	Search	FAR (Hz)	SNR	Labels	Upload Time (UTC)	Added to Superevent (UTC)	Set as Preferred (UTC)
G515079	CBC	pycbc	AllSky	2.462e-09	10.618	SNR_OPTIMIZED PASTRO_READY SKYMAP_READY EMBRIGHT_READY	2024-09-24 00:04:10.80	2024-09-24 00:04:11.74	2024-09-24 00:08:18.38

Figure 7.2: The preferred event for the super event S240924a [? ] identified by the PyCBC Live search pipeline (among others). This PyCBC Live event was created by the SNR optimiser.

5. Keep points with greater network SNR
6. Repeat until the number of max iterations has been reached **OR** the population has converged on a point
7. Output the optimised template solution

These instructions can vary slightly for optimisation algorithms and we will explore these differences as well as the point adjusting algorithms in the following sections.

## 7.2.1 Parameter bounds

The optimiser will be passed the template parameters of the event that triggers the SNR optimisation. From these parameters, the *bounds* are defined. These bounds represent the minimum or maximum value that the optimiser can explore for a specific parameter, preventing it from searching areas we do not want it to search.

The mass parameter bounds are created by converting  $m_1$  and  $m_2$  into chirp mass,  $\mathcal{M}$ , and symmetric mass ratio,  $\eta$ ; this combination leads to a more physically meaningful difference in waveform when changed. The  $\mathcal{M}$  bounds are calculated,

$$\mathcal{M}_{\min} = \begin{cases} \mathcal{M} \cdot \left(1 - \frac{\mathcal{M}}{50.0}\right) & \text{if } \mathcal{M} \cdot \left(1 - \frac{\mathcal{M}}{50.0}\right) \geq 1 \\ 1 & \text{if } \mathcal{M} \cdot \left(1 - \frac{\mathcal{M}}{50.0}\right) < 1 \end{cases} \quad (7.1)$$

$$\mathcal{M}_{\max} = \begin{cases} \mathcal{M} \cdot \left(1 + \frac{\mathcal{M}}{50.0}\right) & \text{if } \mathcal{M} \cdot \left(1 + \frac{\mathcal{M}}{50.0}\right) \leq 80 \\ 80 & \text{if } \mathcal{M} \cdot \left(1 + \frac{\mathcal{M}}{50.0}\right) > 80. \end{cases} \quad (7.2)$$

---

The remaining parameter bounds are set to

$$0.01 \geq \eta \geq 0.2499, \quad (7.3)$$

$$-0.9 \geq s_{1z} \geq 0.9, \quad (7.4)$$

$$-0.9 \geq s_{2z} \geq 0.9, \quad (7.5)$$

$$(7.6)$$

independent of initial event parameters.

Upon creating an instance of the optimiser, a population of candidate solutions is randomly initialised. The first hyperparameter we consider is population size,  $N$ , determining the number of points in the population. Each point is a 4-dimensional vector,  $\mathbf{x}_i$ , given a random value for each parameter inside the bounds,  $\mathbf{x}_i = (\mathcal{M}_i, \eta_i, s_{1z,i}, s_{2z,i})$ , this is our initial population,  $\mathbf{x}_N$ .

### 7.2.2 The original SNR optimiser

The differential evolution optimiser is a stochastic, population-based algorithm that iteratively ‘mutates’ the population to improve individual SNRs. We can visualise the algorithm as a process similar to natural evolution, where, in each iteration, individuals undergo genetic variation through mutation. This combines features from other individuals to create new trial candidates. Just as nature favours the survival of the fittest, the algorithm evaluates the performance of these trial candidates and retains those that exhibit greater SNR, gradually steering the population towards the optimal solution.

When using differential evolution, we have four key hyperparameters:

- **Population size,  $N$ :** Determines how many individuals are in the population.
- **Maximum number of iterations:** Sets how long the algorithm runs, i.e., the maximum number of generations.
- **Mutation factor,  $F$ :** Controls the magnitude of mutation and influences how much individuals are altered by the mutation vectors.
- **Recombination constant,  $R$ :** Also known as the crossover probability, it determines the likelihood that a trial candidate takes parameters from the mutation vector instead of the original individual.

Another hyperparameter, which we do not tune, is the number of processes that are spawned to run the algorithm in parallel. This is set to “as many as possible” when running with PyCBC Live.



The algorithm begins by spawning a population of individuals with initial parameter values within the defined bounds. The network SNR for each individual is calculated, and the individual with the highest SNR,  $\mathbf{x}_{best}$ , is identified. Next, a set of mutation vectors are created,

$$\mathbf{v}_i = \mathbf{x}_{best} + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}), \quad (7.7)$$

where  $\mathbf{x}_{r1}$  and  $\mathbf{x}_{r2}$  are distinct random individuals from the population. A trial candidate,  $u_i$ , is created by combining the mutation vector  $\mathbf{v}_i$  with the target individual  $\mathbf{x}_i$ , according to the rule

$$u_{ij} = \begin{cases} v_{i,j} & \text{if } r_j < R \\ x_{i,j} & \text{otherwise.} \end{cases} \quad (7.8)$$

Here,  $u_{ij}$  is the  $j$ -th parameter of the trial candidate  $u_i$ ,  $v_{i,j}$  is the  $j$ -th parameter of the mutation vector  $\mathbf{v}_i$ , and  $x_{i,j}$  is the  $j$ -th parameter of the target individual  $\mathbf{x}_i$ .  $r_j$  is generated uniformly in the interval  $[0, 1]$ . If  $r_j < R$ , the mutation vector value is used; otherwise, the original target individual's value is retained. One exception is that the final parameter (4th) **always** takes its value from the mutation vector, ensuring that no individual remains completely unchanged in the new population.

Once all trial candidates have been generated, their network SNRs are calculated and compared to the target individual SNR. If a trial candidate has a higher network SNR than the original target individual, it replaces the target. If not, the trial candidate is discarded. The process of creating mutation vectors and generating trial candidates is repeated until the maximum number of iterations is reached, or another stopping criterion is satisfied.

### 7.3 Improvements to the optimiser between observing runs

The PyCBC programme `pycbc_optimize_snr` has been under irregular but constant development to improve the functionality. There are a number of improvements to the programme which have improved the computing time and maximum SNR found. These have been implemented between the third observing run, when the SNR optimiser was created, and the fourth observing run where the SNR optimiser is currently running on live data.



### 7.3.1 Tuning hyperparameters

Tuning the four hyperparameters for the differential evolution optimiser allows for the best performance of the SNR optimiser in the live search. We optimised the following hyperparameters:

- Population size
- Maximum number of iterations
- Mutation factor
- Recombination factor

where a balance must be struck between SNR and computing time. To maximise network SNR we would use larger population sizes, more iterations, higher mutation factors and a lower recombination rate, but this will widen the effective search radius and increase the amount of time taken to find an optimised value.

As shown previously, the original values for the hyperparameters were able to recover the 3% missing SNR from any template mismatch therefore, we chose to tune hyperparameters to reduce the time to run the programme. The computing time is directly related to both the population size and maximum number of iterations, which had initial values  $N = 200$  and `maxiter` = 100. Using a simple grid search over these two hyperparameters for a number of events, we were able to maintain the SNR recovery while reducing the computing time by  $\sim 25\%$  on average using  $N = 100$  and `maxiter` = 50. These were the first set of improvements for the PyCBC Live SNR optimiser.

### 7.3.2 Initial point in the population

The SNR optimiser can miss the maximum SNR of the parameter space if it is not searched with a fine enough set of particles. Not only can the maximum point be missed, but the maximum can be found below the SNR recovered by the initial PyCBC Live event. To prevent this, we can set a floor on the optimiser SNR by including our initial guess into the initial swarm (or population). Including this initial template into the swarm will set the best position, allowing for a faster convergence on the maximum SNR.



### 7.3.3 Astrophysical spin bounds

The bounds described in Section ?? are very general and cover a wide parameters space. The PyCBC Live search is capable of detecting gravitational-wave signals from binary black hole, binary neutron star and, neutron star — black holes. We assume a physical limit on neutron star spin [?] therefore we can impose more astrophysically informed spin bounds on objects with masses consistent with neutron stars

$$\text{spin bounds} = \begin{cases} -0.4 - 0.4 & \text{if max mass} < 3 M_{\odot}, \\ -0.9 - 0.9 & \text{if max mass} > 3 M_{\odot}, \end{cases} \quad (7.9)$$

it is important to impose the condition dependent on the maximum mass the optimiser is allowed to search over. If the optimiser is allowed to assign masses to particles greater than the upper mass limit on neutron stars then we have to let the spin vary by up to the black hole spin bounds because the optimal template could be in this region even if the initial template (from which the bounds are calculated) isn't.

### 7.3.4 Chirp time boundaries

Another improvement made to the parameter bounds is determining the  $\mathcal{M}$  bounds using a constant chirp time window. We define the chirp time as the time between the start of the signal (at some define lower frequency,  $f_L$ ) and the peak frequency prior to the merger

$$\tau_0 = A_0 (\mathcal{M})^{-5/3}. \quad (7.10)$$

where

$$A_0 = \frac{5}{256 (\pi f_L)^{8/3}}. \quad (7.11)$$

The chirp time to Newtonian order,  $\tau_0$ , depends only on the two masses,  $m_1$  and  $m_2$  [?]. By using a chirp time window we can search the  $\mathcal{M}$  parameter space for signals that look far more physically similar than if we were just changing  $\mathcal{M}$  itself.

A higher mass gravitational-wave event will merge at lower frequencies and therefore have a shorter chirp time. To calculate the  $\mathcal{M}$  bounds we first calculate the chirp time of the initial event, then we subtract the constant chirp time window (2 seconds in PyCBC Live) and convert back to  $\mathcal{M}$  to find the maximum  $\mathcal{M}$  bound and add the window and convert to find the minimum  $\mathcal{M}$  bound.

We also improve the  $\eta$  bounds by imposing a minimum mass of any component object  $\min(m) = 0.99$  and calculating the maximum mass of  $m_1$  using the maximum  $\mathcal{M}$ . Using this, we can calculate the *minimum* value of  $\eta$ , again by using the mini-

mum mass of any component and the maximum mass of  $m_1$ . Then using the same maximum  $\eta$  previously defined, we are able to define the upper mass bound on  $m_2$ .

These changes to the parameter bound calculation give us the boundaries for  $\mathcal{M}$  and  $\eta$  as,

$$\left(\frac{A_0}{\tau_0 + 2.0}\right)^{\frac{3}{5}} \geq \mathcal{M} \geq \left(\frac{A_0}{\tau_0 - 2.0}\right)^{\frac{3}{5}}, \quad (7.12)$$

$$\frac{\max(m_1) \cdot 0.99}{(\max(m_1) + 0.99)^2} \geq \eta \geq 0.2499, \quad (7.13)$$

which allow for less time to be wasted in computing regions of the parameter space which we know will not contain the maximum point.

## 7.4 O3 optimisation results

To evaluate the performance of the SNR optimiser in the PyCBC Live search we analyse the events uploaded by the PyCBC Live search to GraceDB. There is a gravitational-wave mock-data challenge (MDC) that runs continuously to enable live search pipelines to test pipelines and changes to the pipelines prior to enabling them on the real gravitational-wave data. The MDC replays 40 days of third observing run gravitational-wave data with simulated injections continuously, providing a much larger rate of signals to test changes on.

We query GraceDB for all PyCBC ‘grace events’<sup>1</sup> uploaded by the PyCBC full bandwidth MDC search between the dates 2024-05-30 and 2024-07-16 and retrieve 10270 grace events, of which 5091 were uploaded by the SNR optimiser. Each grace event has an associated ‘super event’ to which multiple pipelines (and multiple grace events from each pipeline) are assigned, of the SNR optimised grace events, 5008 were assigned a preferred grace event uploaded by PyCBC Live. Finally, of these PyCBC preferred super events, 4600 had SNR optimised grace events as the preferred events. In 91.85% of cases, when the SNR optimiser has uploaded a grace event, it is preferred over the original PyCBC Live uploaded grace event.

Figure ?? shows the change in SNR between original upload and SNR optimised upload for all these SNR optimised events and Figure ?? shows the latency between original upload and SNR optimised upload. The average change in SNR optimised events is a +1.64% increase in SNR. We can see many events which have lost significant amounts of SNR when optimised, these are caused by the bounds of the SNR op-

<sup>1</sup>Grace event refers to an individual event uploaded by a pipeline. The initial PyCBC upload and the SNR optimiser upload are separate grace events.

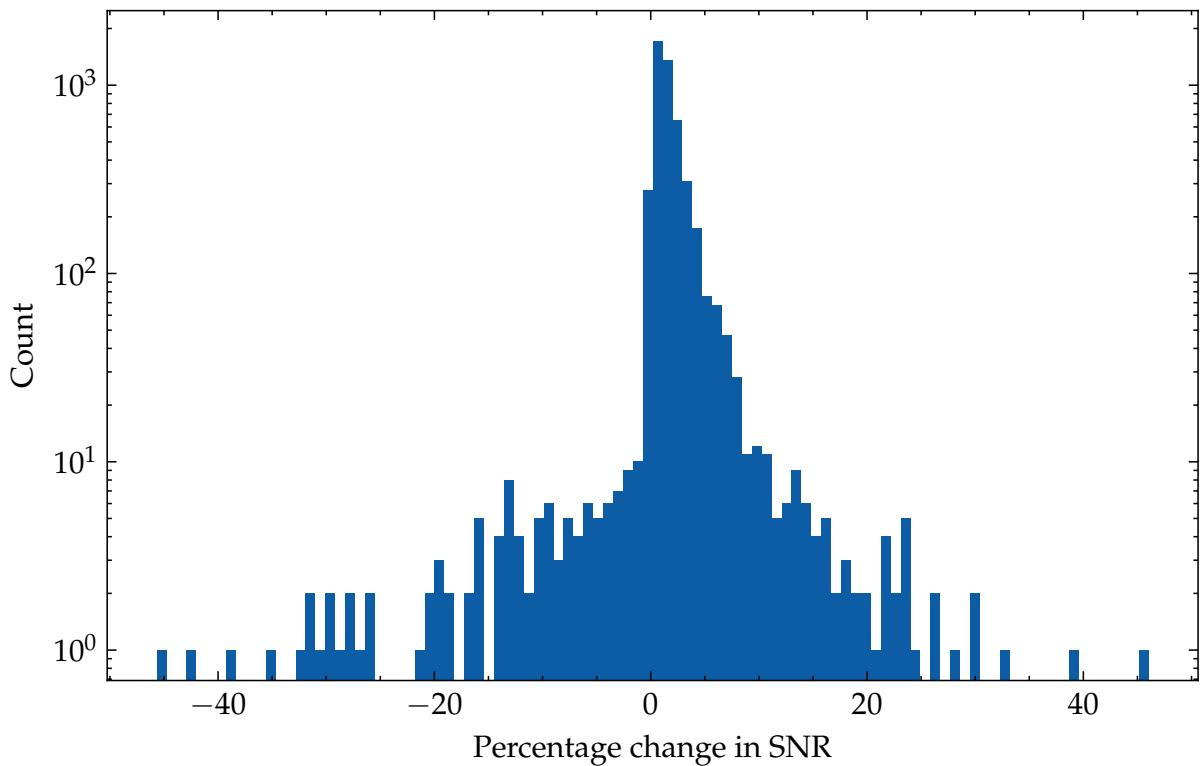


Figure 7.3: The percentage change in SNR between the optimised SNR and original SNR for GraceDB super events found between 2024-05-30 and 2024-07-16 that were found with PyCBC Live and have an SNR optimised grace event in the gravitational-wave data third observing run replay mock-data challenge.

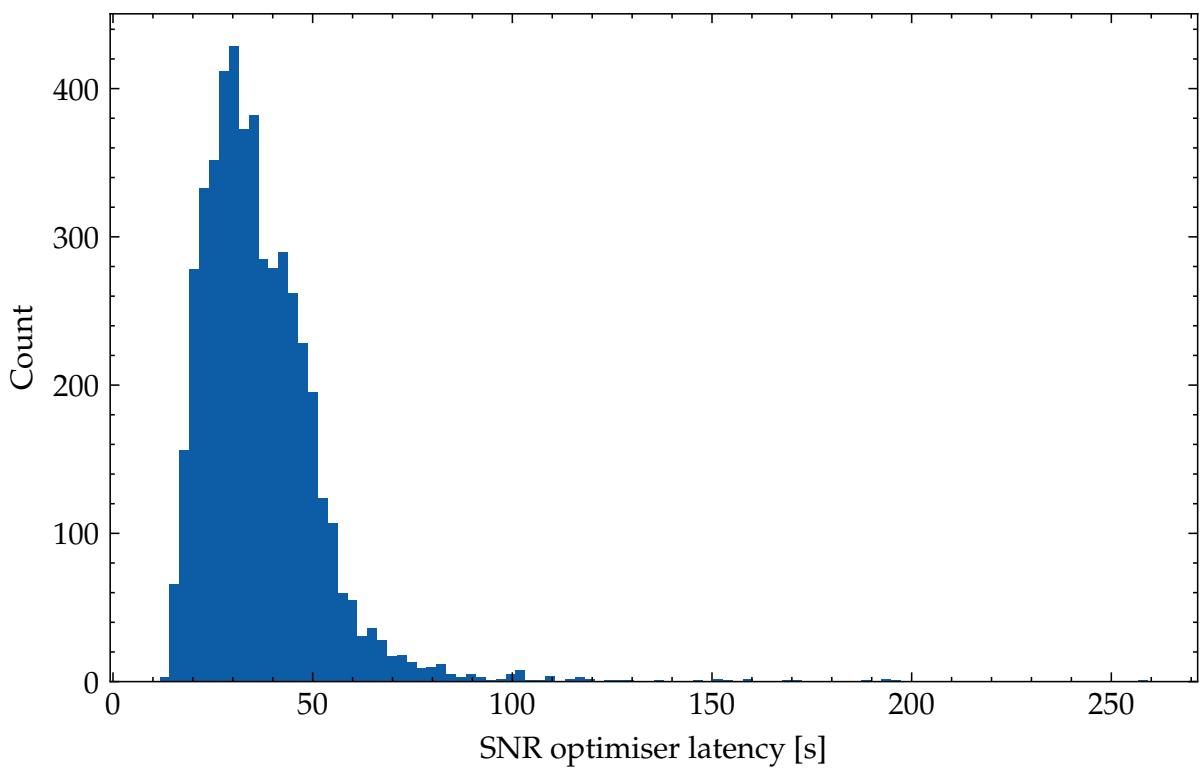


Figure 7.4: The time after original grace event upload for the SNR optimiser to upload an SNR optimised grace event for GraceDB super events found between 2024-05-30 and 2024-07-16 that were found with PyCBC Live and have an SNR optimised grace event in the gravitational-wave data third observing run replay mock-data challenge.

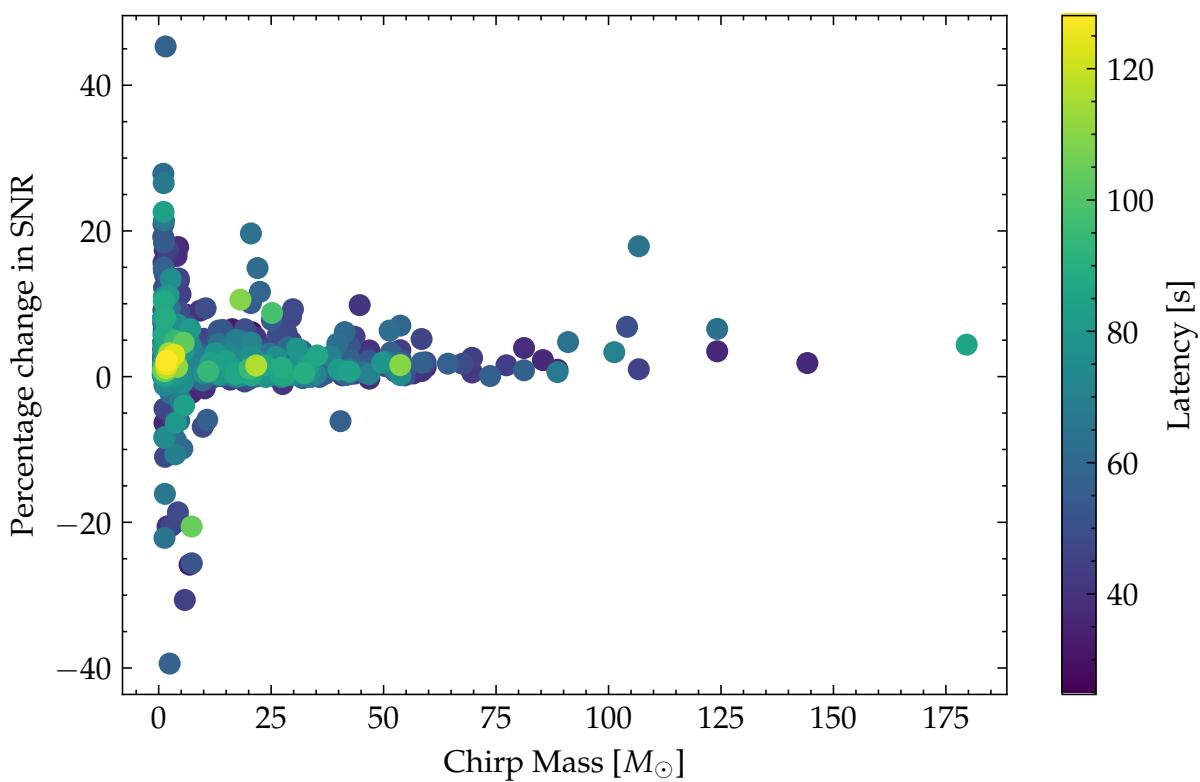


Figure 7.5: The percentage change in SNR against the chirp mass for each gravitational-wave event identified by the MDC search with an SNR optimised event, the injections are coloured by the latency between initial event upload and optimised event upload which is directly related to the time required for the optimiser to run.

timisation parameter space not overlapping with the template bank parameter space or the super event being caused by an exceptional event. As an example, the event with the highest loss in SNR had a negative change in SNR of  $-45.67\%$ , the original grace event parameters were:  $m_1 = 68.91 M_{\odot}$ ,  $m_2 = 1.08 M_{\odot}$ ,  $s_{1z} = 0.994$ ,  $s_{2z} = 0.0$ , this event is very significantly precessing and has a very large mass ratio making the parameter space difficult to optimise over. Even though the original candidate was within the parameter space, the differential evolution optimiser enforces a mutation at every iteration, so the initial best point can be lost and never recovered with the number of iterations being performed. Complicated parameter spaces are difficult for the optimiser to explore in the number of iterations. When the SNR optimiser uploads a grace event, it will achieve an average increase in SNR of  $1.64\%$  with an average latency between uploads of 36.97 seconds. Figure ?? shows the SNR change in optimised events against the event's chirp mass coloured by the latency, it can be clearly seen that lower chirp mass events have longer optimisation times.

## 7.5 Exploring alternate optimisation algorithms

After improving the differential evolution optimisation algorithm, we began to investigate other optimisation algorithms that could be used to greater success for the PyCBC Live problem space. The SciPy package has a module dedicated to optimisers, `SciPy.optimize` which offers 2 alternative optimisers suitable for this optimisation task: brute force optimisation, which involves performing a grid search over the input parameters; and simplicial homology global optimisation (SHGO), an optimisation technique which divides the parameter space into smaller regions and evaluates points in these regions to identify the most promising areas where the best solution might be found [? ].

We dismiss the brute force optimiser immediately and do not consider the SHGO optimiser due to requirements of the parameter space being smooth or non-noisy that we are unsure are fulfilled [? ]. We therefore looked outside SciPy and identified the particle swarm optimisation (PSO) algorithm [? ] as a candidate for the PyCBC Live SNR optimiser. This algorithm has been used for gravitational-wave searches as shown by [? ] and more recently [? ]. This made it a good candidate for optimising network SNR in PyCBC Live.

### 7.5.1 Particle Swarm Optimisation

We can describe the particle swarm optimisation (PSO) algorithm similar to how we previously described the differential evolution algorithm. PSO has the hyperparameters:

- Number of particles,  $N$
- Maximum number of iterations,  $T$
- Inertia weight: weight next iteration velocity based on the previous iteration velocity,  $w$
- Personal learning rate,  $c_1$
- Social learning rate,  $c_2$

the number of particles and maximum number of iterations are identical to the differential evolution hyperparameters. The personal and social learning rates can be considered as acceleration coefficients of the particle.

Once again, we are maximising the network SNR of the PyCBC Live event, adjusting the template parameters ( $m_1$ ,  $m_2$ ,  $s_{1z}$  and  $s_{2z}$ ) to find the optimal values to



give the maximum network SNR. PSO will initialise a “swarm” of particles, each with a unique combination of template parameters which can be considered the position of the particle in the parameter space,  $\mathbf{x}_i$ , the particle will also be given a random velocity,  $\mathbf{v}_i$ . The network SNR of each particle is calculated and the personal best of every particle is set,  $\mathbf{p}_i$  (as this is the only position the particle has occupied). From the swarm, the global best position is determined,  $\mathbf{g}$ .

Next, we update the velocity of each particle

$$\mathbf{v}_{i(t+1)} = w \cdot \mathbf{v}_i(t) + c_1 \cdot r_1 \cdot (\mathbf{p}_i - \mathbf{x}_i(t)) + c_2 \cdot r_2 \cdot (\mathbf{g} - \mathbf{x}_i(t)), \quad (7.14)$$

where  $w$  is the inertial velocity of the particle,  $c_1$  is the personal learning rate,  $c_2$  is the social learning rate,  $r_1$  and  $r_2$  are random numbers uniformly distributed in  $[0, 1]$ ,  $\mathbf{p}_i$  is the personal best position of the particle  $i$  and  $\mathbf{g}$  is the global best position of the swarm. We can break Equation ?? down into three distinct contributions to the velocity of the particle at iteration  $t + 1$ :  $w \cdot \mathbf{v}_i(t)$ , how much weight is placed on the previous velocity of the particle, adjusted using the inertia weight hyperparameter;  $c_1 \cdot r_1 \cdot (\mathbf{p}_i - \mathbf{x}_i(t))$ , how strongly the particle pulls toward its personal best, adjusted using the personal learning weight hyperparameter;  $c_2 \cdot r_2 \cdot (\mathbf{g} - \mathbf{x}_i(t))$ , how strongly the particle pulls toward the global best, adjusted using the social learning weight hyperparameter.

After the velocity of each particle is calculated, the particle positions are updated

$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{v}_i(t + 1), \quad (7.15)$$

and the network SNR of each particle is calculated again. The particle personal bests and the global best are re-evaluated and updated if they have changed, and the iterative process repeats. Upon reaching the number of iterations or if there is convergence in the network SNR value, the algorithm is stopped and the optimal template parameter values are returned which maximise the network SNR.

### 7.5.2 Tuning and evaluating the PSO optimiser

The PSO optimiser has not been used in the PyCBC Live search on either the MDC or real data, but we have implemented it into PyCBC Live to allow the possibility of it being used in the future. We have tested PSO on the Chapter ?? injection set (described in Table ??) which focuses on the low-mass parameter space, the region which the optimiser has struggled to search over.

Using `optuna` [? ], a hyperparameter tuner, we tune the five hyperparameters

---

of the PSO optimiser to achieve the greatest SNR for a single gravitational-wave injection from the injection set. We get rounded hyperparameter values: iterations = 50, particles = 150, inertia weight = 0.2, personal learning rate = 0.75, social learning rate = 1.0. Using these hyperparameter values we search through 21,600 seconds of gravitational-wave data containing gravitational-wave injections every  $\sim$ 100 seconds, with both the current differential evolution optimiser and the PSO optimiser. The differential evolution optimiser achieves an average percent increase in SNR of 0.512% in an average time of 30.489 seconds whereas the PSO optimiser achieves an average increase in SNR of 0.543% in an average time of 32.058 seconds. The PSO optimiser achieves a greater average SNR increase with a slightly longer average optimisation time. We perform the same test and cut the number of iterations to 25, leading to an average optimisation time of 14.638 seconds while maintaining an average SNR increase of 0.543%.

These tests indicate that the PSO optimiser has the potential to replace the differential evolution optimiser in PyCBC Live and with greater hyperparameter tuning we can produce an SNR optimiser which explores the low-mass parameter space effectively.

## 7.6 Conclusion and future improvements

The SNR optimiser will increase the SNR obtained by the PyCBC Live search pipeline for a gravitational-wave event in the vast majority of cases, with an average 1.64% increase. This will lead to more accurate sky maps and improved parameter estimation for the gravitational-wave events. However, as identified by Figure ??, we have a large number of events (8.15%) where the SNR optimiser has **not** increased the found SNR. We have identified that this is true when the parameter space is more complex, in the case of signals with high mass ratio, non-aligned spins or long duration templates. We have a latency requirement on our optimiser and so are unable to necessarily allow the deep exploration of these parameter spaces.

We have identified the particle swarm optimisation algorithm as a candidate for further investigation to improve the PyCBC Live SNR optimiser and we have shown that hyperparameter tuning can lead to improvements in optimised SNR as well as decrease optimisation time therefore requiring further investigation into hyperparameter tuning to improve both the current and future SNR optimiser configurations.

The implementation of the SNR optimiser in PyCBC Live has improved the prospects of gravitational-wave detection in low-latency and will enable higher quality science in the post-detection regime. Providing more accurate sky maps for low-latency



events will allow better multi-messenger follow-up of these events. The SNR optimiser has many areas of potential improvement that if more time were available, we would seek to investigate and implement. In this subsection, I will discuss some of the improvements that could be made to the SNR optimiser.

### Template bank initial population positions

The two optimisers that are currently implemented—differential evolution and particle swarm optimisation—rely on spawning an initial random population of points to explore the template parameter space. We have prior knowledge about the parameter space from our templates bank and can place this initial population of points at the points of our template bank, the points will be drawn toward the best matching point from the template bank and hopefully explore the area between template bank points more deeply. If the parameter space bounds are large for an event this will place the population approximately at the correct density for the parameter space as defined by the stochastic template bank placement algorithm (described in Section ??), more densely covered areas of the template bank will be assigned more initial points to explore that space.

### Template dependent optimisers and hyperparameters

We have adjusted the parameter space bounds according to the initial template parameters, the mass bounds are dependent on the template chirp time and the spin bounds are dependent on the component object (black hole or neutron star). We could apply further options to both optimisers and the hyperparameters of the optimisers, depending on the template. If high-mass BBH signals require fewer iterations we can adjust that hyperparameter, if low-mass signals require a finer exploration of the parameter space we can change the mutation factor and recombination factor or the inertia and learning rates. We might be able to find optimisers that respond better to specific regions of the parameter space, similar to how the template banks are composed using different techniques for different regions.

### Machine learning optimisers and hyperparameter tuning

Since starting this PhD in 2020 the rise of machine learning has been relentless, techniques that were considered novel and cutting-edge have fallen by the wayside and newer, more efficient and effective techniques have replaced them. The SNR optimiser can benefit heavily from this catalogue of machine learning based events that could be highly optimised for an SNR optimisation problem. We have a large number of histor-

ical events to train such a model on to allow enhanced parameter space exploration, which can allow either a faster optimisation or a deeper optimisation, especially for low-mass signals.

Another area in which machine learning can help is with hyperparameter tuning. Optimising the SNR optimiser hyperparameters will return instant reward for the current configuration of the optimiser, the hyperparameters chosen initially or after our initial optimisation may not still be the correct choice in the fourth observing run and in the future. We generate a template bank for each observing run so we should optimise the hyperparameters too.



# 8

---

## Working in Industry

---

*“I can’t wait for you to finish your PhD so I can retire.”*

— Wiktoria Kedziora

As a DISCnet-funded PhD student, I was required to undertake two 3-month placements in industry. These placements were instrumental to my success as a PhD student: the skills, knowledge, and change in environment drove me to learn more in those six months than in some years of my PhD. The placements also helped inform my career decisions and clarified my post-PhD direction.

In this chapter, I will briefly describe the details of my employment and the projects I undertook during these placements. Due to non-disclosure agreements, confidentiality, and the passing of time, I am unable to provide extensive details or specific figures from my work. Section ?? covers my time at Shell Research UK, while Section ?? discusses my work at the Maritime and Coastguard Agency, a UK Government civil service agency.

### 8.1 Shell Research UK

I was hired as a Data Scientist Intern at Shell and placed within the Predictive Maintenance team. The team’s primary focus is to use advanced data science techniques to predict maintenance needs for the many machines at Shell sites globally, helping to prevent unplanned disruptions caused by failing components.



Predictive maintenance aims to anticipate equipment failures by analysing operational data. A machine may refer to anything from a simple pump to a complex refinery system. Components in these machines are fitted with sensors that monitor parameters like temperature, pressure, and vibration. These readings form a high-dimensional dataset reflecting the machine's operational health. For example, a long-term rise in temperature may indicate wear or degradation, signalling an impending failure.

We employed a reconstruction error approach to identify failure patterns. This method involves training machine learning models to understand the normal behaviour of a machine. Reconstruction error measures how closely the model's predictions match actual sensor readings. Significant deviations between predicted and observed values—i.e., a high reconstruction error—can indicate abnormal behaviour and possible component failure.

The machine learning algorithms used in this approach fall under unsupervised learning, which does not rely on labelled data for classification. During my time at Shell, I developed and implemented Principal Component Analysis (PCA) as part of the predictive maintenance pipeline for monitoring machine components.

## 8.2 Maritime and Coastguard Agency

I worked as a Data Scientist Intern at the Maritime and Coastguard Agency (MCA), which is responsible for maritime safety, environmental protection, and coordinating search and rescue operations under the UK Department for Transport.

I was tasked with on-boarding a new Automatic Identification System (AIS) dataset, containing billions of data points tracking every vessel in UK coastal waters over the past four years. This dataset contained numerous errors and anomalies that needed to be processed efficiently and in parallel to ensure the data was usable by other teams within the agency.

At the MCA, I extensively used PySpark—a Python API for Apache Spark that supports large-scale data processing in distributed environments—and Databricks, a cloud platform that provides these environments. With these tools, I developed scalable pipelines to process billions of geospatial data points, impute missing values, calculate new information, and identify errors in the dataset. The unique nature of working with geospatial data allowed me to focus on identifying vessels moving anomalously over land, analysing historical traffic trends in protected areas, and preventing vessel collisions at sea.

# 9

---

## Conclusion

---

*“Re-search? I haven’t found it yet.”*

— Arthur Tolley

We have explored the current methods that have been used to directly detect hundreds of gravitational waves from compact binary coalescences using the most sensitive observatories ever constructed. In this thesis, the research output has focused on improving the search for gravitational wave signals by both developing new tools and applying powerful existing tools in novel ways.

First, we created a novel scattered-light noise artefact model and searched for these noise artefacts, removing them and re-searching for signals to recover previously missed gravitational wave injections. This method was the first of its kind, establishing a glitch search pipeline that employed techniques similar to gravitational wave search pipelines, but tailored to account for properties unique to specific glitch classes. We demonstrated that the approach used in the ArchEnemy pipeline did not significantly improve search sensitivity, but it did show that modelling and removing glitches could allow for the recovery of previously missed gravitational wave injections. In the future, the feasibility of glitch search pipelines could be expanded to include more classes of glitches, enabling a pre-search data cleaning step before gravitational wave detection.

We implemented an exponential noise model in the PyCBC Live search for gravitational waves in low-latency, enhancing the detection ranking statistic used to evaluate



gravitational wave signals. By using a noise model that more accurately describes the characteristics of PyCBC Live’s noise, we demonstrated significant gains in detection sensitivity. While PyCBC Live lacks some ranking statistic components that have been developed for offline gravitational wave searches, future work could convert and optimise these components for PyCBC Live to further improve both the detection ranking statistic and the significance estimation of gravitational wave signals.

Then, we analysed, evaluated, and proposed improvements for the PyCBC Live Early Warning search to maximise the number of gravitational wave events that can be detected in the early warning regime. Running the PyCBC Live Early Warning search over a 40-day mock data set of gravitational wave injections revealed significant deficiencies in the current ranking statistic. Based on this analysis, we suggested increasing the slope of the coincidence timing window allowance and reconfiguring the `phasetd` phase-time-amplitude histograms with specific tuning for early warning. This would allow coincidences to be made when single-detector trigger pairs fall outside the light-travel time. These adjustments could lead to more gravitational wave signals being detected in early warning, and we hope to demonstrate their effectiveness with future binary neutron star events.

We discussed the SNR optimiser used in the PyCBC Live search, as well as improvements to the SNR optimiser algorithm and hyperparameter optimisation performed during this PhD, highlighting the potential for future development in this area. Finally, I discussed my industrial placements during the course of the PhD, where I gained a wealth of knowledge and skills that I brought to my research.

The current epoch of gravitational wave astronomy is ripe for further improvements in search techniques, independent of detector upgrades. New methods can increase gravitational wave search sensitivity in parallel with detector advancements, and these techniques will be applicable to the next generation of detectors, which will explore an even larger gravitational wave parameter space. By enhancing our searches, we can observe more events, and in the case of early warning, provide as much notice as possible, thereby deepening our understanding of the Universe.

# FORM UPR16

## Research Ethics Review Checklist

Please include this completed form as an appendix to your thesis (see the Research Degrees Operational Handbook for more information)



UNIVERSITY OF  
PORTSMOUTH

Postgraduate Research Student (PGRS) Information		Student ID:	839834
PGRS Name:	ARTHUR TOLLEY		
Department:	ICG	First Supervisor:	IAN HARRY
Start Date: (or progression date for Prof Doc students)	2020/10/01		
Study Mode and Route:	Full-time	PhD	

Title of Thesis:	Improving the Detection of Gravitational-Wave Signals in Real Time		
Thesis Word Count: (excluding ancillary data)	38905		

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

### UKRIO Finished Research Checklist:

(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: <https://ukrio.org/publications/code-of-practice-for-research>

a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES
b) Have all contributions to knowledge been acknowledged?	YES
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	YES
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES
e) Does your research comply with all legal, ethical, and contractual requirements?	YES

### Candidate Statement:

I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)

**Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):**

**ETHICS-10374**

If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:

**Signed (PGRS):**

A handwritten signature in black ink, appearing to read "Dr. Michael J. Bolger". The signature is fluid and cursive, with a small oval drawn underneath it.

**Date:** 30/09/2024