University of Trento
Department of Physics
PhD in Physics

~ · ~

Academic Year 2025–2026

Testing the fidelity of waveform reconstruction for gravitational wave transients using minimal assumptions

Supervisor
Prof. Giovanni Andrea Prodi
Co-supervisor
Dott.ssa Claudia Lazzaro

Candidate
Alessandro Martini
240770

Final examination date: July 21, 2026

Don't only practice your art, but force your way into its secrets,
for it and knowledge can raise men to the divine.
Ludwig van Beethoven

Acknowledgments

The author is grateful for computational resources provided by the LIGO Laboratory, supported by National Science Foundation Grants PHY-0757058 and PHY-0823459.This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. This work has been supported by the European Union-Next Generation EU, Mission 4 Component 1 CUP J53D23001550006 with the PRIN Project No. 202275HT58.
This research has made use of data or software obtained from the Gravitational Wave Open Science Center (gwosc.org), a service of the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA.

Abstract

On the 14th of September 2015, the first direct detection of gravitational waves (GWs) was made by the LIGO and Virgo collaborations. This discovery opened a new era in astronomy and physics, allowing us to observe the Universe under a different light. Since then, hundreds of detections have been made in the years, and many more are expected to come. These observations can now be used to test our understanding of the Universe and gain a deeper understanding binding astrophysics, cosmology and General Relativity (GR).
Among all the methodologies to study GW events, burst searches play a central role in the detection of GWs, as they are designed to detect a wide variety of signals without relying on a specific waveform model. Such searches play a crucial role, since they can either detect signals that are not modeled, or be sensitive to deviations from the theoretical predictions of GR. Among all the burst search pipelines, coherent WaveBurst (cWB) is one of the most successful and widely used, as it has been able to detect dozens of signals with high significance.While modeled searches can only reconstruct signals as predicted by GR, burst searches can, in principle, detect deviations from theoretical models, capture hard-to-model distortions of the waveform due to effects like precession or eccentricity, or even help detecting exotic signals like post-merger echoes or cosmic strings—ultimately testing the predictions of GR and looking for new physics.
In such a context, the fidelity of the reconstruction of the waveform plays a crucial role, as only with a good reconstruction of the signal we can perform meaningful tests of the underlying physics.
For this reason, the focus on the thesis is the investigation of the fidelity of the waveform reconstruction inside the burst framework, and in particular with the pythonized version of cWB: py coherent WaveBurst (pycWB).
The first part of the thesis (Chapters 1-4) is dedicated to the introduction of the theoretical background and the state of the art of burst searches, with a particular focus on cWB.
Chapter 1 introduces the theory of GWs in the context of GR, and the history of their detection. Together with this brief introduction, some of the known sources of GWs are introduced. Particular emphasis is put on the sources of GW transients, the main target of burst searches.
Chapter 2 gives a short introduction to the detection of GWs, describing the detectors and the data analysis techniques used in the field.
Chapter 3 is dedicated to the description of the cWB pipeline, with a particular focus on the analysis workflow, the main algorithms and the statistical approach to the detection of candidate signals. Together with the algorithm, its pythonized version (pycWB) is described.
Chapter 4 introduces the principle of Maximum Entropy applied to the estimate of Power Spectral Density of some observed process. This advanced technique displays several advantages with respect to standard ones, providing a low-bias high-resolution estimate of the power spectral density. This is the last theoretical bit needed to understand the results of

the thesis.
The second part of the thesis (Chapters 5-8) is dedicated to the investigation of the fidelity of the waveform reconstruction in the burst framework. The fidelity is assessed by investigating both the bias (Chapters 5 and 6) and the variance (Chapter 7) of the reconstruction. In order, we investigate: the accuracy of the waveform reconstruction (Chapter 5 and 6), the construction of confidence belts around the waveform (Chapter 7), and the construction of a new test statistic based on the concept of the Network Covariance Matrix.
Chapter 5 is dedicated to the investigation of the pre-processing step of the pipeline. We show that two improvements in the pipeline can be achieved. The first regards the tuning of the wavelet transform used in pycWB. We show that the current time-frequency representation is sub-optimal, and that an improved version can be obtained by easily changing the parameters of the transform. The second improvement regards changing the whitening method from the standard one to the Maximum Entropy method described in Chapter 4. We build a new whitening algorithm and compare the pre-processing of the data with the old version of the pipeline and the new one.

Chapter 6 investigates how the improvement of the pipeline made in Chapter 5 affects the accuracy of the waveform reconstruction. Here, we focus on different types of signals, and show that the modification of the wavelet transform leads to a small but detectable improvement in the accuracy of the reconstruction for different signal morphologies.

Chapter 7 is again partly theoretical. The standard cWB and pycWB estimate for the waveform only produce point estimates, i.e. a denoised estimate of the signal over time, with no uncertainty over the process. To address this limitation, we introduce a bootstrap method to estimate the uncertainty of the waveform reconstruction and show how it can be used to construct confidence intervals in the context of pycWB, reporting some case examples.

Chapter 8 takes the improvements made in Chapters 5 and 6 and, using the bootstrap method, and explores a novel method to use burst-estimates to perform statistical inference. We exploit the bootstrap methodology to build a "Network Covariance Matrix", a statistical tool that describes the variability of the reconstruction in a network of detectors, taking into account the correlation between the different detectors and different time samples of every detector. Considering the example of the GW250114 event, we show how this statistic can be used to build a powerful test statistic that can be used in three different cases, namely parameter estimation, model selection and hypothesis testing.

Contents

Chapter 1

Gravitational Waves, Theory and History

1.1 Gravitational Waves Theory

1.1.1 From the Equivalence Principle to Einstein Field Equation

The study of gravitational waves (GWs) originates from their theoretical prediction within the framework of General Relativity (GR), formulated by Albert Einstein in 1915. Central to GR is the Equivalence Principle, a foundational concept in physics with a long and significant history.

Galileo first explored the Equivalence Principle experimentally in the late 16th and early 17th centuries. His experiments demonstrated that objects of different masses fall at the same rate in a gravitational field, suggesting an equivalence between inertial and gravitational mass. This principle was later formalized by Isaac Newton in the late 17th century through his law of universal gravitation, which mathematically described the gravitational force between two masses.

In Newtonian mechanics, the proportionality of inertial and gravitational masses is a natural consequence. Observations and experiments on free-falling bodies necessitate the direct proportionality between these masses:

$$m_i = \alpha m_g \tag{1.1}$$

where α is a constant independent of the object. This formulation is known as the Weak Equivalence Principle (WEP).

Another key concept in the development of GR is the principle of covariance, which requires that the laws of physics maintain the same form in all inertial coordinate systems. This principle was integral to Einstein's formulation of Special Relativity. However, Newtonian gravity could not be expressed in a covariant framework, underscoring the need for a more general theory. The solution came with the introduction of the Strong Equivalence Principle, which extends the WEP to include all reference frames and incorporates the principle of covariance, including non inertial frames. This principle states that all the low of

physics, including gravitational phenomena, must be the same in all reference frames.
The culmination of these ideas allowed Einstein to deduct GR Equations from the above principle. The Einstein Field Equation [84], which is the cornerstone of GR, changes the way we understand gravity in a dramatic way. It describes gravity not as a force, but as curvature of spacetime caused by mass and energy, bonding gravity and geometry. The Einstein Field Equation links with a simple mathematical expression the geometry of spacetime to the distribution of mass and energy within it:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{1.2}$$

with $R_{\mu\nu}$ the Ricci curvature tensor, and is defined by the contraction of the Riemann curvature tensor $R^{\rho}_{\sigma\mu\nu}$ as $R_{\mu\nu} = R^{\rho}_{\mu\rho\nu}$, $g_{\mu\nu}$ the metric tensor, R the Ricci scalar, defined as the contraction of the Ricci tensor $R = g^{\mu\nu}R_{\mu\nu}$, and $T_{\mu\nu}$ the stress-energy tensor, which describes the density and flux of energy and momentum in spacetime.
The Einstein Field Equation is a set of ten interrelated differential equations, which are, in the presence of matter, highly non linear and complex to solve. Non linearity is one of the key features of GR, and it is a direct consequence of the fact that the gravitational field itself carries energy and momentum, which in turn affects the curvature of spacetime. This leads to a feedback loop where the curvature of spacetime influences the motion of matter and energy, which in turn affects the curvature of spacetime.

1.1.2 Linearization of Einstein Field Equation and Gravitational Waves

In this section we see how GWs arise as a natural consequence of GR. A full treatment can be found in [84].
The Einstein Field Equation (1.2) are the equations behind the theory of GR, and they can be used to derive the properties of GWs, as well as the equations that govern their propagation and interaction with matter. The main assumption is that the gravitational field is weak and that the spacetime metric can be expressed as a small perturbation of the Minkowski metric $\eta_{\mu\nu}$, giving

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \tag{1.3}$$

where $\eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$ is the Minkowski metric for the flat space time and $h_{\mu\nu}$ is a small perturbation such that $|h_{\mu\nu}| \ll 1$. By substituting the expression for the metric (1.3) into the Einstein Field Equation (1.2) and linearizing the equations by neglecting terms of order h^2 and higher, we obtain the linearized Einstein Field Equations:

$$\Box\bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4}T_{\mu\nu} \tag{1.4}$$

where $\Box$ is the D'Alembertian operator, defined as $\Box = \partial^{\alpha}\partial_{\alpha} = -\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2$, and $\bar{h}_{\mu\nu}$ is the trace-reversed perturbation defined as

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h \tag{1.5}$$

with $h = h^{\alpha}_{\alpha}$ the trace of the perturbation.
In vacuum, where the stress-energy tensor $T_{\mu\nu} = 0$, the linearized Einstein Field Equations reduce to the homogeneous wave equation:

$$\Box\bar{h}_{\mu\nu} = 0 \tag{1.6}$$

This equation describes the propagation of GWs in free space. For a wave traveling along the z-axis, the solutions to this wave equation are plane waves of the form:

$$\bar{h}_{\mu\nu} = A_{\mu\nu} \cos\left[\omega(t - \frac{z}{c})\right] \tag{1.7}$$

where $A_{\mu\nu}$ is the amplitude tensor, ω is the angular frequency, and c is the speed of light. In vacuum, the gauge and residual gauge freedom can be used to simplify the form of the perturbation, leaving only two physical degrees of freedom, which correspond to the two polarization states of GWs. Under this gauge choice, known as the Transverse-Traceless (TT) gauge, the perturbation $h_{\mu\nu}$ satisfies the conditions:

$$h_{0\mu} = 0, \quad \partial^i h_{ij} = 0, \quad h^i_i = 0 \tag{1.8}$$

where Latin indices run over spatial coordinates.
Considering a wave traveling along the z-axis, in the TT gauge, the amplitude tensor $A_{\mu\nu}$ can be written explicitlely as a function of the two independent polarizations h_+ and $h_\times$ as:

$$h^{TT}_{ij} = \begin{pmatrix} h_+ & h_\times & 0 \\ h_\times & -h_+ & 0 \\ 0 & 0 & 0 \end{pmatrix} \cos\left[\omega\left(t - \frac{z}{c}\right)\right] \tag{1.9}$$

This expression clarifies the name "transverse-traceless", as the perturbation has only spatial components transverse to the direction of wave propagation and the trace of the amplitude tensor is zero.
The two components h_+ and $h_\times$ are the amplitudes of the "plus" and "cross" polarizations, respectively. They describe the way in which the GW stretches and compresses spacetime in different directions as it passes through a region.
This is easily seen by considering the proper distance ds between two events in space time:

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu \tag{1.10}$$

by substituting the perturbed metric (1.3) in the TT gauge with the explicit form of $h_{\mu\nu}$ in Equation (1.8), we obtain

$$\begin{aligned} ds^2 = &- c^2 dt^2 + (1 + h_+ \cos\left[\omega\left(t - \frac{z}{c}\right)\right]) dx^2 \\ &+ (1 - h_+ \cos\left[\omega\left(t - \frac{z}{c}\right)\right]) dy^2 + 2h_\times \cos\left[\omega\left(t - \frac{z}{c}\right)\right] dx dy + dz^2 \end{aligned} \tag{1.11}$$

where we have assumed that the wave is propagating in the z direction. The effect of the GW is to change harmonically the proper distance between two points in space. If we consider two events at $x_1 = (t, x_1, 0, 0)$ and $x_2 = (t, x_2, 0, 0)$ with $L = |x_2 - x_1|$, the proper distance between them is given by

$$s^2 = L^2(1 + h_+ \cos[\omega(t - z/c)]) \tag{1.12}$$

which shows that the proper distance between the two points oscillates with time as the GW passes through. For two events at generic positions in the x-y plane, the proper distance is given by

$$s^2 = L^2 + h_{ij} L^i L^j. \tag{1.13}$$

This is the fundamental principle behind the operation of GW detectors, such as LIGO and Virgo, which measure tiny changes in distance between test masses caused by passing GWs. In fact, the proper distance is related to the time required by a light beam to cross the space,

and its variation can be detected in terms of time of flight. More about this is discussed in Chapter 2.2.

To study in deeper detail the sources of GWs [85], we can start from the linearized Einstein Field Equations 1.7 in the presence of matter. The solution of the equations can be found using the Green's function method, and can be expanded in terms of multipoles. See [84] for a detailed derivation.
From Green's function method, the solution to the wave equation in the presence of matter is given by

$$\bar{h}_{ij}^{TT}(t,\mathbf{x}) = -\frac{1}{r}\frac{4G}{c^4}\Lambda_{ij,kl}(\mathbf{n})\int T^{kl}(t-r/c+\frac{\mathbf{x}'\cdot\mathbf{n}}{c},\mathbf{x}')d^3x' \tag{1.14}$$

where $r = |\mathbf{x}|$ is the distance from the source to the observer, $\mathbf{n} = \frac{\mathbf{x}}{r}$ is the unit vector pointing from the source to the observer, and $\Lambda_{ij,kl}(\mathbf{n})$ is the projection operator onto the transverse-traceless gauge:

$$h_{ij}^{TT} = \Lambda_{ij,kl}\bar{h}_{kl}, \qquad \Lambda_{ij,kl} = P_{ik}P_{jl} - \frac{1}{2}P_{ij}P_{kl}; \qquad P_{ij} = \delta_{ij} - n_i n_j \tag{1.15}$$

In the far-field limit, where the distance from the source to the observer is much larger than the size of the source, we can approximate the integral by evaluating the stress-energy tensor at the retarded time $t - r/c$. This gives

$$\bar{h}_{ij}^{TT}(t,\mathbf{x}) = -\frac{1}{r}\frac{4G}{c^4}\Lambda_{ij,kl}(\mathbf{n})\int T^{kl}(t-r/c,\mathbf{x}')d^3x' \tag{1.16}$$

To proceed further, we can expand the stress-energy tensor in terms of multipoles. The leading order term in this expansion is the quadrupole moment, which is given by

$$Q_{ij}(t) = \int \rho(t,\mathbf{x})(x_i x_j - \frac{1}{3}\delta_{ij}r^2)d^3x \tag{1.17}$$

where $\rho(t,\mathbf{x})$ is the mass density of the source. The quadrupole moment describes the distribution of mass in the source and how it changes with time.
The GW strain h_{ij}^{TT} can then be expressed in terms of the second time derivative of the quadrupole moment as

$$h_{ij}^{TT}(t,\mathbf{x}) = \frac{2G}{c^4 r}\frac{d^2Q_{ij}^{TT}(t-r/c)}{dt^2} \tag{1.18}$$

where Q_{ij}^{TT} is the transverse-traceless part of the quadrupole moment.
This equation shows that GWs are generated by the time-varying quadrupole moment of a mass distribution. The above equation holds for any source of GWs, as long as the weak-field approximation is valid. This means that we are neglecting the back-reaction of the GWs on the source itself. At the same time, the above equation is valid only in the far-field limit, where the distance from the source to the observer is much larger than the size of the source and, last but not least, for non-relativistic sources, where the velocities of the masses in the source are much smaller than the speed of light.
For non-relativistic sources, the multipole-expansion is equivalent to an expansion in powers of v/c, where v is the typical velocity of the masses in the source. This means that the quadrupole formula is valid only for sources where the velocities are much smaller than the speed of light, and implies that only systems with a non-zero second time derivative of the quadrupole moment can emit GWs. This includes systems such as binary star systems, supernovae, and rotating neutron stars with asymmetries.

Notice that the there is a factor of G/c^4 in front of the equation, which makes the amplitude of GWs extremely small. This is why GWs are so difficult to detect, and why it took so long to observe them directly.
In most cases, the distribution of mass is needed to compute the quadrupole moment and its time derivatives, making it very difficult to obtain an exact solution. However, in some cases, such as binary systems, we can make some approximations to obtain an estimate of the GW strain.

1.2 History of Gravitational Waves Detection

1.2.1 First evidence of Gravitational Waves: The Hulse-Taylor Binary Pulsar

The publication of the theory of GR and GWs were met with skepticism by the scientific community. The scientists were doubtful about the physical reality of various aspects of the theory, which became established only after several experimental confirmations. The first evidence of the validity of GR came from the prediction of the perihelion precession of Mercury's orbit, which was observed and measured with high precision. Another important confirmation came from the deflection of light by the Sun, which was observed during a solar eclipse in 1919. However, the existence of GWs remained a topic of debate for several decades, with most scientists considering them as a mere mathematical artifact of the theory. The first indirect evidence of GWs came from the observation of the Hulse-Taylor binary pulsar [66], PSR B1913+16, discovered in 1974 by Russell Hulse and Joseph Taylor. This system consists of two neutron stars orbiting each other, one of which is a pulsar emitting regular radio pulses. By measuring the timing of the pulses from the pulsar, Hulse and Taylor were able to determine the orbital parameters of the system with high precision. They found that the orbit of the pulsar was decaying over time, which was consistent with the emission of GWs predicted by GR. The rate of orbital decay was found to be in excellent agreement with the predictions of GR, providing strong indirect evidence for the existence of GWs. For their discovery, Hulse and Taylor were awarded the Nobel Prize in Physics in 1993.

The Hulse-Taylor binary pulsar remains one of the most important tests of GR and the existence of GWs, as it provided the first experimental confirmation of GR and paved the way for further research in the field of GW astronomy. It was only several decades after the discovery of the Hulse–Taylor binary pulsar that gravitational waves were first detected directly. This required the development of highly sensitive detectors, such as the LIGO-Virgo-KAGRA (LVK) [1, 16, 27] interferometers, which require a combination of advanced technology and sophisticated data analysis techniques. It is only thanks to these technological advancements that the first direct detection of GWs has been made in 2015, as discussed in Chapter 2.2.

1.3 Main sources of Gravitational Waves

Gravitational waves can be generated by an extremely wide range of both astrophysical and cosmological sources. For our purposes, we divide the type of sources in three major groups [101]:

- Continuous Waves
- Stochastic Background
- Gravitational Waves Transients

The first two categories consist of persistent signals, whose durations can extend beyond the detector network's total observation time. Continuous waves (CWs) are long lasting and theoretically modeled, usually quasi-monochromatic signals, emitted by rotating neutron stars with some asymmetry. The Stochastic Background is a superposition of a large number of unresolved sources, both of astrophysical and cosmological origin, that results in a random and persistent GW signal. Each of these categories of signals is targeted by specific search techniques and data analysis methods, which are designed to efficiently extract the signal from the noise in the detector data.
In contrast, GW transients are short-lived, and this thesis focuses exclusively on short bursts with durations on the order of $\mathcal{O}(1\text{ s})$ or shorter. These transient signals can be divided into two distinct classes: Compact Binary Coalescences (CBCs) and burst signals. The primary distinction between these two groups lies in the level of theoretical modeling and characterization available for each class. The CBC signals are well constrained and behaved, and their morphology is well predicted by GR by a set of parameters describing the binary and its position in the sky, namely the intrinsic and extrinsic parameters of the source. The CBCs are well-modeled, so that they can be analyzed using techniques that leverage *a priori* knowledge of the expected signal morphology. This includes the use of matched-filtering methods for signal detection or Bayesian inference frameworks to extract the physical parameters of the source [118]. On the contrary, Burst signals are poorly constrained by theoretical models or, even worse, they can exhibit a stochastic behavior or being originated from unpredicted sources. They can be both short living and long living, but we only focus on the first scenario throughout this thesis. A typical example for this are the GWs emitted during a Core-Collapse Supernovae (CCSNe), where the complex dynamics of the collapse and the extreme conditions make it very difficult to predict the exact waveform of the emitted GWs. Several models of CCSNe have been developed, but the exact waveform of the emitted GWs remains uncertain and is expect to be inherently stochastic [9]. In the following, we provide a brief overview of each category of sources.

1.3.1 Continuous Waves

Continuous waves are long-lasting and "well-defined", usually quasi-monochromatic signals, emitted by the rotation of a neutron star around an axis not aligned to its symmetry axis. This is not the only possible source for CWs, but it is the most relevant for ground-based detectors, such as LIGO and Virgo. These signals are expected to be long lasting, and should be present in the data continuously over the observation time. The CW can be modeled by its quadrupole moment, which is related to the moment of inertia of the neutron star by:

$$Q_{ij} = I_{ij} - \frac{1}{3}\delta_{ij} I \tag{1.19}$$

where I_{ij} is the moment of inertia tensor of the neutron star, and I is its trace. The constant trace term can be neglected, as it does not contribute to the GW emission (which only depends on the second time derivative of the quadrupole moment as shown in Eq 1.18). In the reference frame $\Sigma' : (x'_1, x'_2, x'_3)$ of the neutron star, taking the rotation axis to be

along x_3', the moment of inertia tensor can be expressed as

$$I'_{ij} = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \tag{1.20}$$

where I_1, I_2, and I_3 are the principal moments of inertia. If the neutron star is rotating around the x_3' axis with angular frequency Ω, one can re-write the moment of inertia tensor in the reference frame $\Sigma : (x_1, x_2, x_3)$ of the observer by a simple rotation of the reference frame:

$$I_{ij} = R_{ik} R_{jl} I'_{kl} \tag{1.21}$$

where R_{ij} is the rotation matrix given by

$$R_{ij} = \begin{pmatrix} \cos\omega t & -\sin\omega t & 0 \\ \sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.22}$$

In this reference, the various component of the moment of inertia tensor can be expressed as

$$\begin{aligned} I_{11} &= I_1 \cos^2\omega t + I_2 \sin^2\omega t \\ &= \frac{I_1 + I_2}{2} + \frac{I_1 - I_2}{2}\cos 2\omega t \end{aligned} \tag{1.23}$$

$$\begin{aligned} I_{22} &= I_1 \sin^2\omega t + I_2 \cos^2\omega t \\ &= \frac{I_1 + I_2}{2} - \frac{I_1 - I_2}{2}\cos 2\omega t \end{aligned} \tag{1.24}$$

$$I_{12} = (I_1 - I_2)\sin\omega t \cos\omega t = I_{21}$$

The other components are $I_{13} = I_{23} = 0$, and $I_{33} = I_3$, hence all constants that do not contribute to the GW emission.
We notice that the only time-varying components of the moment of inertia tensor are I_{11}, I_{22}, and I_{12}. As for the case of the CBCs, the GW frequency is twice the rotation frequency of the neutron star, $f_{GW} = 2f_{\rm rot}$, and the time-dependent term are all proportional to the difference between the two principal moments of inertia $I_1 - I_2$. This means that the amplitude of the GW emission from a rotating neutron star depends on the degree of asymmetry in the mass distribution, which can be quantified by the ellipticity ϵ defined as

$$\epsilon = \frac{I_1 - I_2}{I_3}. \tag{1.25}$$

and expressed in a more compact form, defining the characteristic amplitude of the GW signal as

$$\begin{aligned} h_+ &= h_0 \frac{1 + \cos^2\iota}{2}\cos(2\pi f_{GW} t) \\ h_\times &= h_0 \cos\iota \sin(2\pi f_{GW} t) \end{aligned} \tag{1.26}$$

where ι is the inclination angle between the rotation axis of the neutron star and the line of sight to the observer, and

$$h_0 = \frac{4\pi^2 G I_3 f_{GW}^2}{c^4 r_{\rm obs}}\epsilon \tag{1.27}$$

such that a typical amplitude for a rotating neutron start, with a mass $m \simeq 1.4M$ and a radius $a \simeq 10km$, which gives $I_3 \simeq (2/5)ma^2 \simeq 1 \times 10^{38}\text{kg m}^2$, at a 10Kpc distance and ellipticity of 10^{-6}, is given by

$$h_0 = 1.06 \times 10^{-25} \left(\frac{I_3}{10^{38}\text{g cm}^2} \right) \left(\frac{f_{GW}}{1\text{kHz}} \right)^2 \left(\frac{10\text{kpc}}{r_{\text{obs}}} \right) \left(\frac{\epsilon}{10^{-6}} \right) \tag{1.28}$$

Making them a challenging target for current ground-based detectors. Notice that the amplitude scales with the square of the frequency, making higher-frequency neutron stars more promising sources for CWs. The most recent advancement are reported in [10, 11]

1.3.2 Stochastic Background

Another type of GW sources is the Stochastic Background, which is a superposition of a large number of unresolved sources. The sources can be both of astrophysical and cosmological origin, resulting in a random and persistent GW signal.
The astrophysical contribution to the stochastic background arises from a variety of sources, including CBCs, rotating neutron stars and supernovae that, for their large number and/or their distance, cannot be individually resolved by current detector, but their combined emission can produce a detectable background signal.
The cosmological contribution to the stochastic background is thought to originate from processes in the early universe, such as inflation, phase transitions, and cosmic strings. These processes can generate GWs with a broad range of frequencies, potentially contributing to the stochastic background observed today.
On of the most interesting techniques to characterize the GW Stochastic Background is using Pulsar Timing Arrays [64, 65]. PTAs are networks of millisecond pulsars that are monitored for variations in their pulse arrival times. GWs passing between the Earth and the pulsars can induce correlated timing variations, allowing for the detection of low-frequency GWs in the nanohertz range, either resolving single sources or measuring the statistical properties of the GW background.

1.3.3 Gravitational Waves Transients

The type of sources described in the previous sections are long-lasting and persistent, and require ad hoc analysis for their detection. For our discussion, we focus on a last category of sources, which are the GWs "Transients". These sources are characterized by their short duration in the detector band, and is currently below $\sim 1s$. We further divide this category in two sub-categories: CBCs and "Burst" sources. The main difference between these two groups is the amount of knowledge and characterisation that we have, from the theory, for the two signals. CBC signals are theoretically well modelled using GR, and their detection and characterization is well performed using data analysis methods [118] that rely on the explicit characterization of the source model. On the contrary, "Burst" sources are poorly constrained by theoretical models, can exhibit stochastic behavior, or can be originated for non-anticipated sources, and they are addressed with ad-hoc data analysis techniques called unmodeled (or burst) searches. While CBCs can be targeted by both methodologies, the Burst sources can only be searched with unmodeled searches. The difference between these two searches are discussed in Chapter 2.2.

Compact Binary Coalescences

One of the most import sources of GWs are the CBCs [95, 96], which are systems of two compact objects, such as black holes (BH) or neutron stars (NS), orbiting around each other. Such system emit GWs due to the time-varying quadrupole moment of the mass distribution. In this case, the quadrupole moment can be expressed in terms of the masses of the two objects, m_1 and m_2, and their separation distance r. For such systems, in the first phase of the inspiral, where the two objects are far apart and moving slowly, we can use the quadrupole formula to estimate the GW strain. In the center of mass frame, the quadrupole moment is given by

$$Q_{ij} = \mu(x_i x_j - \frac{1}{3}\delta_{ij} r^2) \tag{1.29}$$

where $\mu = \frac{m_1 m_2}{m_1+m_2}$ is the reduced mass of the system, and x_i are the coordinates of the relative position of the two objects.
The second time derivative of the quadrupole moment can be computed using the equations of motion for the two-body system, which can be approximated by Newton's law of gravitation. For a circular orbit with orbital frequency ω, after applying the transverse-traceless projection, the two polarization components of the GW strain in the quadrupole approximation are:

$$h_+(t) = \frac{4G^2\mu M}{c^4 r_{\rm obs}} \frac{1}{r^2}(1 + \cos^2\iota)\cos(2\omega t) \tag{1.30}$$

$$h_\times(t) = \frac{8G^2\mu M}{c^4 r_{\rm obs}} \frac{1}{r^2}\cos\iota \sin(2\omega t) \tag{1.31}$$

where M is the total mass of the system, $r_{\rm obs}$ is the distance from the source to the observer and ι is the inclination angle of the orbital plane with respect to the line of sight. Note that the GW frequency is twice the orbital frequency, $f_{GW} = 2f_{\rm orb}$, reflecting the quadrupolar nature of the emission.
The amplitude of the GW scales as $h \sim \frac{G^2\mu M}{c^4 r_{\rm obs} r^2}$, showing the dependence on the masses, separation distance, and distance to the observer.

Evolution of Compact Binary Systems: Inspiral, Merger, and Ringdown

The evolution of a compact binary system can be divided into three distinct phases: inspiral, merger, and ringdown.
Inspiral Phase: During the inspiral phase, the two objects orbit each other at relatively large separations. As they emit GWs, they lose energy and angular momentum, causing the orbit to decay. The orbital frequency increases as the separation decreases, following the relation

$$\frac{df}{dt} = \frac{96\pi}{5}\left(\frac{G\mathcal{M}}{c^3}\right)^{5/3}(2\pi f)^{11/3} \tag{1.32}$$

where $\mathcal{M} = \mu^{3/5}M^{2/5}$ is the chirp mass, a combination of the component masses that determines the rate of frequency evolution. This phase produces a characteristic "chirp" signal, with increasing frequency and amplitude. The quadrupole approximation and post-Newtonian corrections provide accurate predictions for the waveform in this phase.
Merger Phase: As the two objects approach each other, the separation becomes comparable to their Schwarzschild radii, and the weak-field approximation breaks down. The objects plunge toward each other and merge, forming a single compact object. This phase is highly

non-linear and relativistic, requiring full numerical relativity simulations to accurately model the GW emission. The merger produces the strongest GW signal, with maximum amplitude occurring at the peak of the merger.
Ringdown Phase: After the merger, the newly formed object (typically a black hole) is highly distorted and undergoes damped oscillations as it settles into a stable configuration. These oscillations, called quasi-normal modes, emit GWs with characteristic frequencies and damping times determined by the mass and spin of the final black hole. The ringdown signal can be modeled as a sum of exponentially damped sinusoids:

$$h(t) \propto \sum_n A_n e^{-t/\tau_n} \cos(\omega_n t + \phi_n) \tag{1.33}$$

where ω_n and τ_n are the frequency and damping time of the n-th mode. The ringdown phase provides information about the properties of the final black hole and serves as a test of GR in the strong-field regime.
The complete GW signal from a CBC event contains information from all three phases, allowing us to extract the masses, spins, and distance of the system, as well as testing fundamental physics in extreme gravitational environments. CBC signals are the primary targets for ground-based GW detectors like LIGO and Virgo, which have successfully observed numerous such events since 2015[41]. Most of these searches are based on matched filtering techniques, which require accurate waveform models for the entire inspiral-merger-ringdown sequence, but several detections have also been made using unmodeled search techniques[45].

Deviations for templates

These models are well constrained in the case of quasi-circular orbits with negligible eccentricity and no precession. If the system is actually representative of these conditions, Bayesian inference provides the best framework to study these sources. However, templated searches constrain the signal to some specific model, and deviation in the physical source from the assumption cannot be retrieved by the search, leading to a decrease in the detection efficiency and/or bias in the inferred parameters. In these context, burst searches can be relevant to detect and characterize such deviations.

Eccentricity

One of the most important deviation from the expected CBC signals is a non-negligible Eccentricity in the orbit of the two compact object[42]. The standard CBC templates banks are built under the assumption of quasi-circular orbits, which is a reasonable assumption for isolated binary systems, as the emission of GWs tends to circularize the orbit over time. However, some dynamic formation scenarios can lead to the formation of binaries with significant eccentricity up to the merger time, and can be a signature of the formation channel of the binary system.
Nevertheless, eccentricity is hard to model and it is not included in the current template banks [104]. This can significantly impact the searches for CBC signals in the high eccentricity limit, leading to a decrease the detection efficiency or bias the inferred parameters [60]. It has been shown that the sensitivity of the modelled searches decreases for eccentricity values above $e \sim 0.1$ at a false alarm rate (FAR) of both $1/10 year^{-1}$ and $1/100 year^{-1}$ [99], while the efficiency of the unmodeled search is not affected by the presence of eccentricity in the signal. This makes eccentric waveforms a possible target for burst searches.

Memory

Another interesting effect that can lead to deviation from the expected CBC signal is the presence of a non-linear memory effect[36]. The memory effect is a non-linear phenomenon in GR, where the passage of a GW can cause a permanent change in the spacetime metric, resulting in a lasting displacement of test masses.
Memory is related to the non-linear nature of GR and is a signature of the larger asymptotic symmetry group of GR compared to e.g. special relativity.
It has been shown that the inclusion of such an effect in the searches, if existent, in the high SNR regime ($SNR > 90$) would help breaking the Distance-Inclination degeneracy while its neglection could lead to systematics in the estimated values of distance and inclination angles [124].

Deviations from GR

Burst searches can be a competitive approach to detect and characterize deviations from the expected CBC signals. CBCs are an ideal target to test GR in the strong-field regime. Alternative theories can predict deviation from the expected CBC signal, such as the presence of additional polarizations [34], or the existence of Exotic compact objects (ECOs). The ECOs are example of possible deviations from GR at the level of the Planck Scale, for which quantum mechanical effects can lead to the formation of object that differ from black holes at the horizon scale, which can lead to the repeated emission of GWs after the merger[120]. There is no unique consensus on the exact morphology of such deviations, so that burst searches [91] are needed to detect and characterize such signals.

Core-Collapse Supernovae

Core-collapse supernovae [116, 14] are another important source of GWs. For their properties, they are the archetype of burst sources. CCSNs are the terminating phase of the life of a massive star whose mass if above $\sim 8M_{\odot}$. During the collapse, the core of the star implodes under its own gravity, leading to the formation of a neutron star or a black hole. The energy of the GW emission from a CCSN is expected to be much smaller than that of a CBC, making them more difficult to detect. Galactic CCSNs are expected to produce GW signals with amplitudes of the order of $h \sim 10^{-21}$ at a distance of 10 kpc, which is within the sensitivity range of current detectors for nearby events. However, the rate of CCSNs in our galaxy is relatively low, estimated to be about 2-3 per century [19], making the detection of such events a rare occurrence.

The CCSNs are example of multi-messenger sources, as they emit electromagnetic radiation, neutrinos and GWs. The observation of either the neutrinos or the gravitational waves would be extremely relevant in understanding the dynamic of the collapse. In fact, both the neutrinos and the GWs are emitted from the innermost region of the star, shortly after the collapse and have a very small interaction with the matter [67], making them ideal messengers to probe the dynamics of the collapse and the properties of the collapse mechanism.

Unfortunately, there is one non-electromagnetic observation of a CCSN so far, which is the observation of neutrinos from SN 1987A [62]. The detection of neutrinos from a CCSN provides valuable information about the dynamics of the collapse and the properties of the progenitor star. GWs from a CCSN would provide valuable information about the dynamics of the collapse, the properties of the progenitor star, and the mechanism of explosion.

The exact waveform of the emitted GWs remains uncertain, and there are several numerical models of CCSNs that predict different waveforms, depending on the specific dynamics of the collapse and the properties of the progenitor star. The GW emitted by a CCSN is expected to consist of two main phase, an initial bounce which halts the initial collapse of the core, and a second stochastic component whose spectral features evolve in time. The models are based on different assumptions and approximations, and can lead to different predictions for the amplitude, frequency, and duration of the emitted GWs. These uncertainty over the emission model make the CCSNs a perfect target for unmodeled searches, and they are currently targeted with All-Sky generic searches[38] or via targeted searches following up electromagnetic observations of CCSNe in the nearby universe, as the recent detection of the CCSN 2023ixf [4].

Neutron Star Glitches

Neutron Star (NS) glitches [23] are sudden increase of the spin frequency of the neutron star. The glitch consists of two main phases, the glitch itself, which is the sudden increase of the spin frequency, usually in the order of milliseconds, and the recovery phase, which is the gradual return to the pre-glitch spin frequency. The main mechanism behind the glitch is thought to be the transfer of angular momentum from the superfluid to the crust, but the quantitative details of the mechanism are still not well understood.
Glitches are phenomena that carry information about the internal structure of the neutron star and can be relevant in understanding the physics of matter at extreme densities. They are also expected to be sources of GWs, as the sudden change in the spin frequency can lead to a time-varying quadrupole moment. Studying this signal can help us to constrain the nuclear matter equation of state, which is one of the main open question in the physics of neutron stars.
The GW is related to the f-mode excitation of the neutron star, and the expected frequency of the emitted GWs is in the range of 2-3 kHz and a damping time of the order of 10ms to .5ms s [63], depending on mass and the equation of state for the neutron star. The minimum detectable frequency change for a glitch in O4a (the first part of the fourth observing run of the LVK collaboration) is of order $\Delta\nu/\nu \sim 10^{-5}$. As for other Sources NS glitches can be targeted by follow-up searches after the observation of a glitch in the electromagnetic spectrum, as for the recent Vela Glitch in 2024 [39], or with all-sky generic searches[38].

Chapter 2

Gravitational Waves Detection, A Modern Perspective

2.1 Introduction

Having discussed the theoretical aspects of GWs in the previous chapter, together with some of the possible sources, we now move our focus to the experimental side of the field, for the detection of GWs. The goal of this chapter is to give a brief overview of the GW detectors are their working principles.

2.2 The Interferometers

Gravitational Wave detectors are complex, large scale interferometers that are designed to measure with high accuracy the distortion of space time cause by the passage of a GW. The successful observations of GWs achieved since 2015 have all been possible thanks to the network of interferometers of the LVK collaboration. The basic working principle of these detectors is based on the Michelson interferometer, where a laser beam is split into two perpendicular arms, reflected back by mirrors, and then recombined to produce an interference pattern. The key, physical principle behind the detection is the change in the proper distance between two points caused by the passage of a GW. As discussed in the previous chapter, the passage of a GW in the 'z' direction causes a periodic stretching of the space-time on the orthogonal plane, which can be described as a variation of the space-time metric as described in Equation 1.11, so that the two polarizations act independently on the 'x' and 'y' directions for the plus polarization, and on the 'x' and 'y' directions together for the cross polarization. The change in the proper distance is than proportional to the amplitude of the GW, and is of order

$$\frac{\Delta L}{L} \sim h \sim 10^{-21}. \tag{2.1}$$

Such a small variation in length can only be detected with extremely sensitive instrument, and the idea of a Michelson interferometer (with several, complex adaptations) allows to detect such small variation of the distances. The scheme of the Michelson interferometer for the Virgo detectors is reported in Fig. 2.1. The most simplified version, to understand the physical principle behind the detection, can be considered following the path of the laser

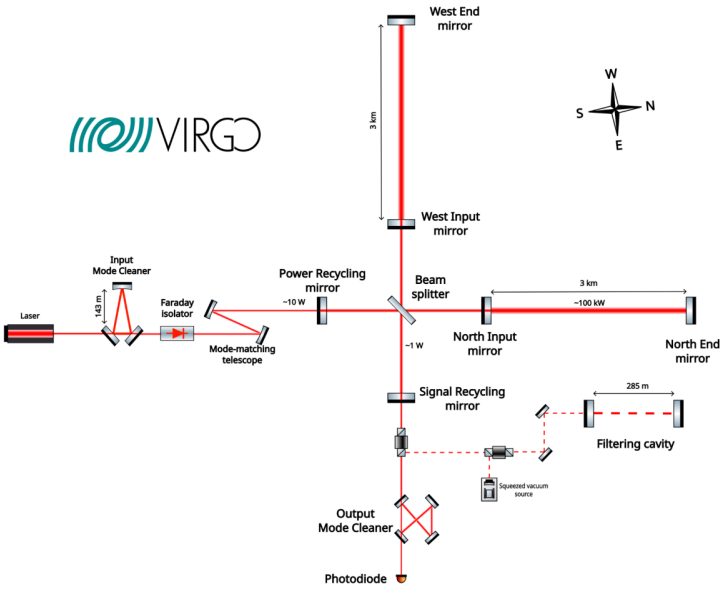


Figure 2.1: Scheme of the Virgo interferometer, courtesy of the Virgo Collaboration.

beam into the interferometer up until the beam splitter. At this point the beam is split into two and sent to the two arms cavities of the interferometers, where the beam is reflected by the End mirror and sent back to the beam splitter, where it is recombined and gets sent to the photo-detector. At this level the other components of the interferometer are not needed to understand the basic principle.
We can consider the case of a monochromatic laser beam with frequency ω_L and wave vector k_L. The electric field of the beam can be written As

$$E(t, \mathbf{x}) = E_0 e^{-\imath(\omega_L t - \mathbf{k}_L \mathbf{x})} \tag{2.2}$$

with E_0 the amplitude of the electric field. If the two arms of the interferometer have different lengths, the divided beams enter the arms simultaneously but exit them at different times, introducing a relative time delay before they recombine at the photodetector. When recombined at the photodetector, the energy of the beam is defined as [84]:

$$|E_0|^2 = E_0^2 \sin^2 [k_L(L_y - L_x)] \tag{2.3}$$

with L_x and L_y the lengths of the two arms. In the ideal case, when the two arms have the same length, the beams will recombine destructively at the photo-detector, and no light will be detected. Any variation in the length of the arms will cause a variation of the power at the photo-detector. In the TT-Gauge, we can treat the mirrors as free falling masses, and the passage of the GW does not cause a change in the coordinate distance between the mirrors, but it affects the travel of the light from the beam splitter to the mirrors and back. Since the mirrors do not change position, in the TT-Gauge the passage of the GW induces an overall phase shift in the light beam, that in the limit $L \ll \lambda_{GW}$, with λ_{GW} the wavelength of the GW, can be written as:

$$\Delta\phi \simeq k_L L h(t - L/c), \tag{2.4}$$

which is formally equivalent to

$$\frac{\Delta(L_x - L_y)}{L} \simeq h(t - L/c). \tag{2.5}$$

The overall effect in the displacement of the mirrors is directly proportional to the total length of the arms, so that long arms are necessary to detect small variation of the length. We stated that the above results are obtained in the limit L much smaller than the wavelength of the GW λ_{GW}, so that L also plays a role in the limit of the detectable wavelength of the interferometer. If the wavelength of the GW is smaller than the length of the arms, the phase shift induced by the passage of the GW will be averaged out over the length of the arms, and the sensitivity of the interferometer will decrease. One can show that the optimal length to maximize the sensitivity to a certain frequency is

$$L \simeq 750\,\mathrm{km}\left(\frac{100\,\mathrm{Hz}}{f_{gw}}\right), \tag{2.6}$$

which is way larger than the actual length of the arms of the current interferometers, that are of about 4 kms for the two LIGO detectors and 3 kms for the Virgo detector. Such long arms are impossible to build on Earth, so that some workarounds to maximize the sensitivity of the interferometer are needed. To increase the effective length of the arms, the current interferometers use Fabry-Perot cavities in the arms, in which the beam is reflected back and forth. When the cavity is at resonance, the light beam interferes constructively with itself, and the cavity becomes extremely sensitive to tiny variation of the length of the arms. The other components of the interferometer are:

- Power recycling mirror: it is placed before the beam splitter, and it reflects back the light that would be lost at the input of the interferometer, increasing the power of the laser beam in the interferometer.
- Signal recycling mirror: it is placed after the beam splitter, and it reflects back the light that would be lost at the output of the interferometer, increasing the sensitivity of the interferometer to certain frequencies.
- Input mode cleaner: it is placed before the power recycling mirror, and it is used to filter the laser beam, removing any unwanted frequency component and spatial mode.
- Output mode cleaner: it is placed after the signal recycling mirror, and it is used to filter the output beam, removing any unwanted frequency component and spatial mode.

2.2.1 Interaction with Gravitational Waves

Having described its working principle, we can now briefly discuss how the interferometer interacts with the passage of a GW. Since the detection is only possible thanks to the relative change in the length of the arms caused by the passage of the GW, we can already conclude that whatever source that acts equally on the two arms will not be detected, since it will not cause any of the power at the photodetector.
The sensitivity of the detector to a GW depends on the direction of the source with respect to the interferometer and it is described by the antenna patterns, F_+ and $F_\times$, that are defined as the response of the interferometer to a GW with plus and cross polarization,

respectively. The antenna pattern for LVK detectors with orthogonal arms are defined in the proper detector frame as:

$$F_+ = \frac{1}{2}(1 + \cos^2\theta)\cos 2\phi \tag{2.7}$$

$$F_\times = \cos\theta \sin 2\phi, \tag{2.8}$$

where θ and ϕ are the polar and azimuthal angles, respectively, and the GW signal detected by the interferometer takes the form:

$$h(t) = F_+ h_+(t) + F_\times h_\times(t). \tag{2.9}$$

As we can see, there are certain directions in the sky for which the interferometer is completely blind to the passage of a GW. For waves with a single polarization, the total number of blind spots increases. For an elliptically polarized wave (i.e. the two polarizations have different amplitudes and are out of phase), the interferometer is blind to the passage of the wave only for sources located at $\cos 2\phi = 0$ and $\cos\theta = 0$, which means $\phi = \pm\pi/4$ and $\theta = \pm\pi/2$ and maximum sensivity found at the north and south poles of the sky in the detector frame.

2.2.2 Sources of noise

As we have seen in the previous section, the interferometers are designed to detect extremely small variation of the length of the arms, to detect the passage of a GW, causing the variation of the total power of recombined beam at the photo detector.
Together with the signal, there are several different phenomena that cause a variation of the power, and they act as sources of noise for the detection and the data outputted by the interferometer. The output of the interferometer (our data) will then be a time series given by the superimposition of the signal and the noise,

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

with $h(t)$ the detector response to the signal 2.9 and $n(t)$ the noise. Since the noise is what prevents us from detecting the signal, its characterization is crucial to understand the sensitivity of the instrument and to perform the data analysis. Since $n(t)$ is a stochastic process, the only way to characterize it is purely statistical, i.e. it is fully described by its distribution $p(n(t))$.
The distribution of noise plays a crucial role, since it can be shown that the distribution of the data is given by the convolution of the distribution of the signal and the distribution of the noise, so that, for a deterministic signal $h(t)$, we have

$$p(d(t)) = p(n(t))\Big|_{n(t)=d(t)-h(t)} \tag{2.11}$$

The distribution for the data is the distribution of the noise computed on the residuals of the data, defined as the difference between the data and the signal (yet to be detected). The distribution of the noise is then crucial to understand the Likelihood of the data and to perform statistical inference.
The distribution for the noise can be complex, but it can be simplified by making some assumption. In particular, some common assumptions are that the distribution for the noise is Gaussian and wide-sense stationary (i.e. the mean and autocorrelation function of the

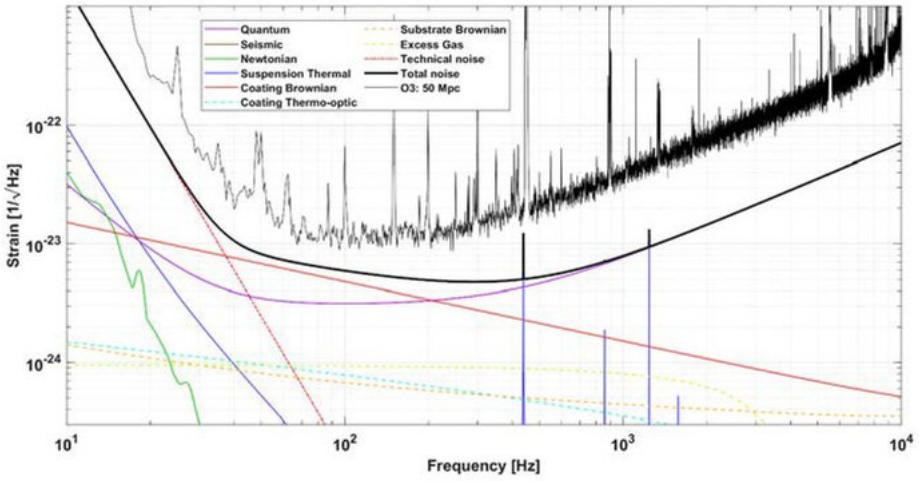


Figure 2.2: Noise Amplitude spectral density of the virgo detector [46].

process do not depend on time); under these assumptions, in the limit of infinite data, the distribution for the noise take a simple form in the Fourier space:

$$p(n) \propto \exp\left[-\frac{1}{2}\int_{-\infty}^{+\infty} df \frac{|\tilde{n}(f)|^2}{S_n(f)}\right], \tag{2.12}$$

with $\tilde{n}(f)$ the Fourier transform of the noise and $S_n(f)$ the power spectral density (PSD) of noise. Under this assumption, the PSD is the only quantity needed to fully characterize the noise and, together with it, the sensitivity of the instrument[1].

Figure 2.2 shows the amplitude spectral density (ASD), i.e. the square root of the PSD (PSD), for the Virgo detector, together with its main noise contributions. There are several different sources of noise, but the main ones are:

- **Seismic noise**: it is caused by the seismic activity of the Earth, and the waves can propagate through the ground causing the mirrors to move. It can be reduced by using seismic isolation systems in the suspension [46].

- **Thermal noise**: it is caused by the thermal motion of the atoms in the mirrors coatings and the suspension system, and it is the dominant source of noise at intermediate frequencies (between 10 Hz and 100 Hz). It can be reduced by using low-loss materials for the mirrors and the suspension system, and by cooling the mirrors to cryogenic temperatures, as KAGRA is attempting to do [27]. The normal modes of the mirrors annd the suspension system also cause resonances in the noise spectrum, that can be seen as peaks in the ASD.

[1]In practice, real gravitational-wave detector data frequently violates these assumptions. Non-stationarity manifests as long-term drifts in the detector's sensitivity, while non-Gaussianity is primarily introduced by instrumental or environmental artifacts. To handle non-stationarities, data analysis pipelines estimate the PSD, in segments adjacent to the one being analyzed (For instance this work in chapter 5.4.2 propose an unsupervised learning algorithm to take these effects into account). The non-Gaussian excursions are addressed either by data gating (zeroing out the affected data segments) or by simultaneously modeling and subtracting the glitches via algorithms like `BayesWave` [48].

- **Quantum noise**: Quantum noise has two different contributions:
  - *Shot noise*: it is caused by the quantum nature of light, and it is the dominant source of noise at high frequencies (above 100 Hz). The shot noise is Poissonian noise related to the fact that the number of photons detected at the photodetector is a random variable, and its variation is actually measured at the photodetector. Since it is Poissonian, the relative fluctuation of the number of photons is given by $\sqrt{N}/N$, so that to reduce the shot noise, we need to increase the power of the laser beam in the interferometer.
  - *Radiation pressure noise*: It is caused by the force exerted by the photons on the mirrors, and is proportional to the power of the laser beam in the interferometer. It is dominant at low frequencies and can be reduced by using heavier mirrors or light squeezing techniques [2, 18].
- **Newtonian noise**: it is caused by the gravitational interaction between the mirrors and the surrounding environment, such as the seismic activity of the Earth, the atmospheric pressure variations, and the human activity. It is dominant at low frequencies and can be reduced by using underground detectors and by monitoring the environment around the detectors.

There are additional sources of noise, but we do not delve into them in detail. All these noise sources contribute to the overall noise of the interferometer, and their effects are encoded in the distribution. However, there are other types of noise that cannot be modeled in as a 'Gaussian' distribution, and they are known as "glitches". Glitches are short-duration, non-Gaussian noise transients that act similarly to a signal than to the stochastic noise. Their impact on the analysis is non trivial, and they can affect the analysis in different ways, decreasing the statistical significance of a detection and biasing the parameter estimation of the source [100, 94, 82].
The way to mitigate the presence of glitches is twofold. Ideally, one can try to understand their physical origin, by monitoring the environment around the detectors and use auxiliary channels for their detection and mitigation [77, 51]. Unfortunately this is not possible for all the sources of glitches, so that some ad hoc procedures are needed to detected and characterize them directly in the data using machine learning techniques [125] or time-frequency analysis [102].

2.3 Data Analysis, short overview

The last problem in the detection of GWs is the data analysis, which, from a high-level perspective, may be interpreted as disentangling the signal from the noise in the data, i.e. finding the best estimate $\hat{h}(t)$ for $h(t)$. There are several ways to proceed with the data analysis, but they can be categorized in two main classes: modeled (templated) and unmodeled searches. The key principle on which they differ is that modeled searches rely on strong assumptions on the signal, and are grounded on a theoretical model which, in this case, derives from GR. Conversely, unmodeled searches rely on minimal assumptions about the signal and are inherently data-driven.
The difference between these two approaches is then reflected in the choice of the methodologies and tools used to perform the data analysis, ending up with completely different results. The work of this thesis is focused on the unmodeled (or burst) searches, and specifically on coherent WaveBurst (cWB) [56], and its new python implementation. In this section instead, we give a brief overview of the templated search in the context of short transients,

such as those produced by CBCs.
The data analysis can be divided into two main steps: the detection of a signal, which consists in finding a signal in the data and assessing its statistical significance, and the interpretation, that in the case of modeled searches consists in the estimation of the parameters of the source.

2.3.1 Signal Detection

The first step consists in the detection of a signal in the data, i.e. discerning from two alternative hypothesis in which $\mathcal{H}_0$ is the null hypothesis of noise only, i.e. $d(t) = n(t)$, and $\mathcal{H}_1$ is the alternative hypothesis of signal plus noise, i.e. $d(t) = h(t) + n(t)$. In the case of GWs, in general, $h(t) << n(t)$, so that the signal is buried in the noise and its detection is non-trivial.
The idea behind the detection is to filter the data in such a way that the power of the signal relative to the noise (the SNR) is maximized. A filter K acts on the data d as

$$\hat{d} = \int_{-\infty}^{+\infty} dt d(t) K(t) \tag{2.13}$$

The signal to noise ratios (S / N) is defined as the ratio of the amplitude of d when the signal is present to the amplitude of d when only noise is present, i.e. The energy of d is computed as the expectation on the filtered data, i.e.

$$S = \int_{-\infty}^{+\infty} dt \mathbb{E}[d(t)] K(t) = \int_{-\infty}^{+\infty} dt \mathbb{E}[h(t) + n(t)] K(t) = \int_{-\infty}^{+\infty} dt h(t) K(t), \tag{2.14}$$

since $\mathbb{E}[n(t)] = 0$ by definition. Note that this expectation operator reflects a theoretical property of the assumed probability distribution for the ensemble, rather than an empirical average over the observed time series. The energy of the noise is computed as the variance of the noise process, i.e.

$$N^2 = \mathbb{E}[\hat{d}^2] - \mathbb{E}[\hat{d}]^2 = \int_{-\infty}^{+\infty} dt \int_{-\infty}^{+\infty} dt' K(t) K(t') \mathbb{E}[n(t) n(t')] \tag{2.15}$$

that, in the Fourier space can be written as:

$$N^2 = \int_{-\infty}^{+\infty} df |\tilde{K}(f)|^2 S_n(f), \tag{2.16}$$

with $S_n(f) = \mathbb{E}[\tilde{n}(f)\tilde{n}^*(f)]$ the noise PSD. The SNR is then given by the ratio of the two,

$$\frac{S}{N} = \frac{\int_{-\infty}^{+\infty} dt h(t) K(t)}{\sqrt{\int_{-\infty}^{+\infty} df |\tilde{K}(f)|^2 S_n(f)}}. \tag{2.17}$$

It is possible to show [84] that the optimal filter, defined as the filter that maximized the SNR, is given by

$$\tilde{K}(f) = C \frac{\tilde{h}(f)}{S_n(f)}, \tag{2.18}$$

with C some constant. We can see that the optimal filter can only be computed if both the PSD and the signal are known. The PSD is not derived from the theoretical noise

models discussed in the previous section; rather, it is directly estimated from the data using established computational methods [79, 121]. Conversely, the expected signal is typically modeled using the theoretical framework of GR. For CBCs the data are filtered using a template bank, i.e. a collection of waveforms that covers the parameter space of the possible sources [50]. These templates are applied to the data using the matched filter, and the SNR is computed for each template. If the SNR exceeds a certain background-dependent threshold, the template is considered as a candidate detection.

The SNR statistics

If the noise was purely gaussian, i.e. if it was completely described by the distribution in 2.12, the SNR would be a random variable with a chi-squared distribution with 2 degrees of freedom [84], so that the statistical significance of a detection could be easily computed by comparing the SNR of the candidate detection with the distribution of the SNR under the null hypothesis. However, as we have seen in the previous section, the noise of the interferometers is affected by glitches and other non-stationarities that modify the distribution of the SNR. In fact, the glitches are found to happen often in the detector at relatively high values for the SNR. In the first part of O4 run glitches were found to happen at a rate of $53.7/h$ for triggers with SNR above 6.5 and $30.8/h$ for SNR > 10 in the LIGO detectors [108], inflating the distribution of the SNR and affecting the estimate of the statistical significance of the candidates.
One of the methods to take into account the modifications to the statistical significance of the candidates is to use the "time-shift" method, explained in detail in 3.3.1.

2.3.2 Parameter Estimation

Having detected a signal in the data, after assessing its statistical significance, the next step is the interpretation, i.e. finding the set of parameters that, assumed GR, best describe the signal. This is done with Bayesian inference and Posterior Maximization [117, 6, 111].
Given a model M for the signals and a set of parameters θ that describe the model, we can write the posterior distribution for the parameters as

$$P(\theta|d, M) = \frac{\mathcal{L}(d|\theta, M)p(\theta|M)}{\pi(d|M)}, \tag{2.19}$$

where $\mathcal{L}(d|\theta, M)$ is the likelihood, $p(\theta|M)$ is the prior and $\pi(d|M)$ is the evidence, which is relevant in model selection. , The likelihood is usually computed using the normality assumption [57], On the left hand side of the equation there is the Posterior distribution for the parameters P, which encodes all the knowledge about the parameters after observing the data. It represents the best state of knowledge over the set of parameters describing the observation. Having defined the Posterior distribution, the best set of parameters (their estimate θ_0) is described as the point in the parameter space that maximizes the posterior distribution, called the Maximum A Posteriori (MAP) estimate.
Sampling from the posterior is a complex task, especially when the parameter space is high-dimensional, such as in the case of the CBCs, in which the total number of parameters can be up to 15 when intrinsic and extrinsic parameters are considered. To perform the sampling, different techniques are used, such as Markov Chain Monte Carlo (MCMC) methods [117, 25] and Nested Sampling [107, 53, 61, 75].

Chapter 3

Coherent Wave Burst 2G

When searching for GWs, several approaches exist and are adopted in research. Two main approaches can be distinguished. The first approach is based on the use of "templated" searches. As discussed in the previous chapter, these searches are based on prior knowledge of the expected signal and aim to match the data with a template of the expected signal. The second approach is based on the use of "unmodeled" searches: these searches do not rely on prior knowledge of the expected signal but instead attempt to identify a signal in the data by detecting a "common" excess power in a network of detectors. The unmodeled searches are also referred to as "Burst" searches, and Coherent WaveBurst 2G (cWB 2G) [56, 71] is one of the most successful software tools used for this purpose. In this chapter, we introduce cWB 2G, its main features, and the key algorithms. In the first section, we discuss the Wilson-Daubechies-Mayer transform (WDM), which is one of the core concepts behind cWB. The WDM transform is a wavelet transform used to represent the data in a manner suitable for coherent analysis. We then introduce the complete workflow of cWB, with particular emphasis on two steps of the analysis: **whitening** and **Likelihood**. Whitening plays a central role in this thesis, and its impact on the analysis, along with other possible implementations, is the focus of the second part of the thesis. Likelihood, on the other hand, is crucial for waveform reconstruction, which is the focus of the third part of the thesis and serves as one of the benchmarks for testing the whitening process.
At the end we also introduce py-coherent WaveBurst (pycWB) [123, 122]. Initially developed as a Python wrapper for cWB 2G, pycWB has evolved into a pipeline that implements the same algorithms as cWB but is entirely written in Python. The core principle are, at the moment of writing, shared between the two implementations, so that at this stage the discussion is relevant for both pipelines. All the results of this thesis from chapter 5 on, are obtained using pycWB.

3.1 The Wilson-Daubechies-Mayer Transform

Before introducing the cWB 2G workflow, we discuss the WDM [92, 47], a fundamental concept underlying the cWB 2G. The WDM is a wavelet transform that converts data from a pure time-domain representation to a time-frequency (TF) representation, which is particularly suitable for burst analysis. Taken a sample time-series $x[t] = \{x[1], x[2], \ldots, x[N]\}$ of N points sampled at a fixed interval Δt and duration $T = N\Delta t$, the WDM is a wavelet transform that allows to represent any such time-series as a linear combination of basis

functions $g_{jk}[t]$ defined in both time and frequency:

$$x[t] = \sum_{j=1}^{N_t} \sum_{k=1}^{N_f} \omega_{jk} g_{jk}[t]. \tag{3.1}$$

In this representation the time-series $x(t)$ is represented by a rectangular grid, which consists of a set of coefficients ω_{jk}, also known as "pixels". The pixels represent the signal in both time and frequency. The index j is the time index, spanning on the N_t "time bins" of size ΔT, while k is the frequency index spanning on the N_f "frequency bins" with size ΔF. The size of the bins are defined as

$$\Delta T = N_f \Delta t \tag{3.2}$$

$$\Delta F = \frac{1}{2\Delta t N_f}, \tag{3.3}$$

and the area of each pixel is a constant equal to $\Delta T \Delta F = \frac{1}{2}$. The number of frequency bins N_f is a free parameter and is defined by the "resolution level" of the transform L. Switching from one resolution level L to the next one $L+1$ corresponds to a multiplication of N_f by a factor of two. This implies that, moving from one resolution level to the following, the time resolution ΔT doubles, while the frequency resolution ΔF is halved. As $N_f \to 1$, the transform approaches the original time series, while as $N_f \to N$, the transform approaches the Fourier Series.
The $g_{jk}[t]$ are the basis functions of the WDM transform, which are defined in both time and frequency. They form a complete orthonormal basis, satisfying the orthonormality condition:

$$\langle g_{jk}, g_{j'k'} \rangle = \delta_{jj'} \delta_{kk'}. \tag{3.4}$$

Using the orthogonality condition in equation 3.4, the WDM coefficients ω_{jk} are computed as:

$$\omega_{jk} = \sum_{t=1}^{N} x[t] g_{jk}[t]. \tag{3.5}$$

Equations from 3.1 to 3.5 hold for any complete orthonormal wavelet basis. What distinguishes the WDM is the specific construction of the basis functions $g_{jk}[t]$. In the frequency domain, the WDM basis functions are defined as:

$$\tilde{g}(\omega)_{n0} = e^{-\imath n \omega T} \tilde{\phi}(\omega), \tag{3.6}$$

$$\tilde{g}(\omega)_{nm} = \frac{1}{\sqrt{2}} e^{-\imath n \omega T} \left(C^*_{nm} \tilde{\phi}(\omega + m\Delta\Omega) + C_{nm} \tilde{\phi}(\omega - m\Delta\Omega) \right), \tag{3.7}$$

and the coefficients C_{nm} are defined as:

$$C_{nm} = \begin{cases} 1 & \text{if } n+m \text{ is even,} \\ \imath & \text{if } n+m \text{ is odd.} \end{cases} \tag{3.8}$$

The function $\tilde{\phi}(\omega)$ is the Meyer window function [90], defined as:

$$\tilde{\phi}(\omega) = \begin{cases} \frac{1}{\sqrt{\Delta\Omega}} & \text{if } |\omega| < |A|, \\ \frac{1}{\sqrt{\Delta\Omega}} \cos\left(\frac{\pi}{2} \nu_\beta(\omega) \left(\frac{|\omega| - A}{B} \right) \right) & \text{if } A < |\omega| < A + B, \end{cases} \tag{3.9}$$

To fully specify the WDM transform, we need to define the parameters A, B, and β. The parameter A defines the frequency range where the window function is constant, spanning from $\omega = -A$ to $\omega = A$. Thus, A can be interpreted as half the frequency width of the constant region. The parameter B, instead, defines the frequency range where the window function transitions smoothly. These parameters are constrained by $2A+B = \Delta\Omega$. In cWB, the parameters are set such that $2A = B = \frac{\Delta\Omega}{2}$, meaning half of the frequency band has a constant window function, while the other half transitions smoothly. The smoothness of the transition outside the interval $[-\frac{\Delta\omega}{4}, \frac{\Delta\omega}{4}]$ is controlled by the normalized incomplete Beta function $\nu_\beta(\omega)$:

$$\nu_\beta(x) = \frac{\int_0^x y^{\beta-1}(1-y)^{\beta-1}dy}{\int_0^1 y^{\beta-1}(1-y)^{\beta-1}dy}. \tag{3.10}$$

The parameter β controls the smoothness of the window function, affecting its frequency resolution. Larger β values result in smoother window function in the frequency domain. The standard value of β in cWB is $\beta = 6$, which provides finest frequency resolution. Different β values are discussed in Chapter 5.1. The Meyer window function is represented in Figure 3.1 for the value $\beta = 6$ at three different resolution levels. The right plot shows the Meyer window in the frequency domain (equation 3.9), while the left plot shows the time envelope, obtained via the inverse Fourier transform. From the figure it is evident that the resolution level acts as a scaling factor of 2 for the two domains.

By inserting the definition of the basis functions $g_{jk}[t]$ 3.6 into equation 3.5, one can show that [49]

$$\omega_{jk} = \sqrt{2}\Delta t \,\mathrm{Re}\left\{ C_{nm} \sum_{k=-K/2}^{K/2-1} e^{2\pi\imath kmq/K} x[nN_f + k]\phi[k] \right\}, \tag{3.11}$$

with $K = 2qN_f$ the width of the window function and $q > 1$ a parameter that sets the size of the window function. By the above definition, one can recognize that the summation is equivalent to a diescrete Fourier transform of $x[nN_f + k]$ multiplied by the Meyer window function $\phi[k]$, i.e.

$$X[mq] = \sum_{k=-K/2}^{K/2-1} e^{2\pi\imath kmq/K} x[nN_f + k]\phi[k]. \tag{3.12}$$

the projection onto the basis g_{jk} can be efficiently computed using a Fast Fourier Transform (FFT) of the original time series.
The last element that characterizes a wavelet transform is how far it lays from the the Gabor uncertainty limit [59]. Having defined the basis functions, one can compute the uncertainty σ of the transform in time and frequency as:

$$\sigma_t^2 = \int_{-\infty}^{\infty} (t-\bar{t})^2 |\phi(t)|^2 dt, \quad \sigma_f^2 = \int_{-\infty}^{\infty} (f-\bar{f})^2 |\phi(f)|^2 df. \tag{3.13}$$

Gabors uncertainty principle states that the product of the uncertainty in time and frequency is bounded

$$C = \sigma_t \cdot \sigma_f \geq \frac{1}{4\pi} \tag{3.14}$$

Any wavelet transform has its own "tradeoff" between the time and frequency resolution, i.e. $\sigma_t\sigma_f = C$, with C a constant that depends on the specific transform. The Gabor

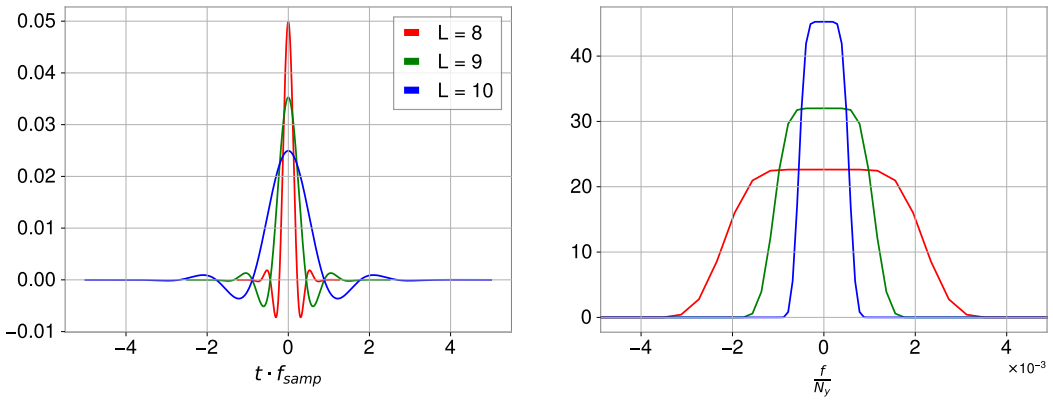


Figure 3.1: The WDM envelopes in time (left) and frequency (right) as a function of the resolution level. A larger resolution level improves frequency resolution and decreases time resolution, and vice versa.

uncertainty principle implies that an improved resolution in one domain leads to a worse resolution in the other domain. The resolution level is a perfect example of this tradeoff, as increasing the resolution level improves the frequency resolution while worsening the time resolution, and vice versa. For every resolution level, the constant C of the transform is unchanged, since it only depends on the specific form of the basis functions. In fact, L only acts as a scaling factor for the basis functions, and does not change their shape. The value for the parameter β instead, since it changes the shape of the basis functions, can change the value of C. This effect is studied in Chapter 5.2. The Gabor uncertainity principle poses a fundamental limit on the capability of the transform to resolve signals in both time and frequency, and any transform has its own trade-off. For example, if we wanted to distinguish spectral components that are very close in frequency, we would need to increase the frequency resolution, of the transform by changing, for example, the resolution level. But acting in this way, it means that we won't be able to distinguish signals that are very close in time, i.e. whose time distance Dt is $Dt \leq \frac{C}{\sigma_f}$. Conversely, if we wanted to distinguish signals that are very close in time, we wouldn't be able to distinguish frequency components that are very close in frequency. The Gabor uncertainty limit of $C = \frac{1}{4\pi}$ is the best trade-off between time and frequency resolution, and no transform can do better than this limit. Figure 3.2 shows the result of the WDM transform of a zero-noise CBC signal of equal masses $M_1 = M_2 = 40M_\odot$ at a sampling frequency of 4096 Hz and different resolution. The resolution levels go from $L = 3$ up to $L = 10$. The CBC signal evolves both in time and frequency and is perfect to illustrate the trade-off between the time and the frequency resolution. At low resolution levels, the frequency resolution is poor and different frequency components are averaged inside the very same bin. The time evolution in instead clearly captured. On the opposite, at high resolution level, the frequency components are highly resolved, but the time evolution is lost, due to the long duration of the time bins. A good representation of the evolution is obtained at intermediate resolution levels, from 6 to 8, where the chirp structure of CBC signals is clearly visible.
Since different features of the signal are captured by different resolution levels cWB analysis are not performed at a single resolution level, but it employs a multi-resolution approach, combining together different resolution level to capture as many features as possible in the two domains.

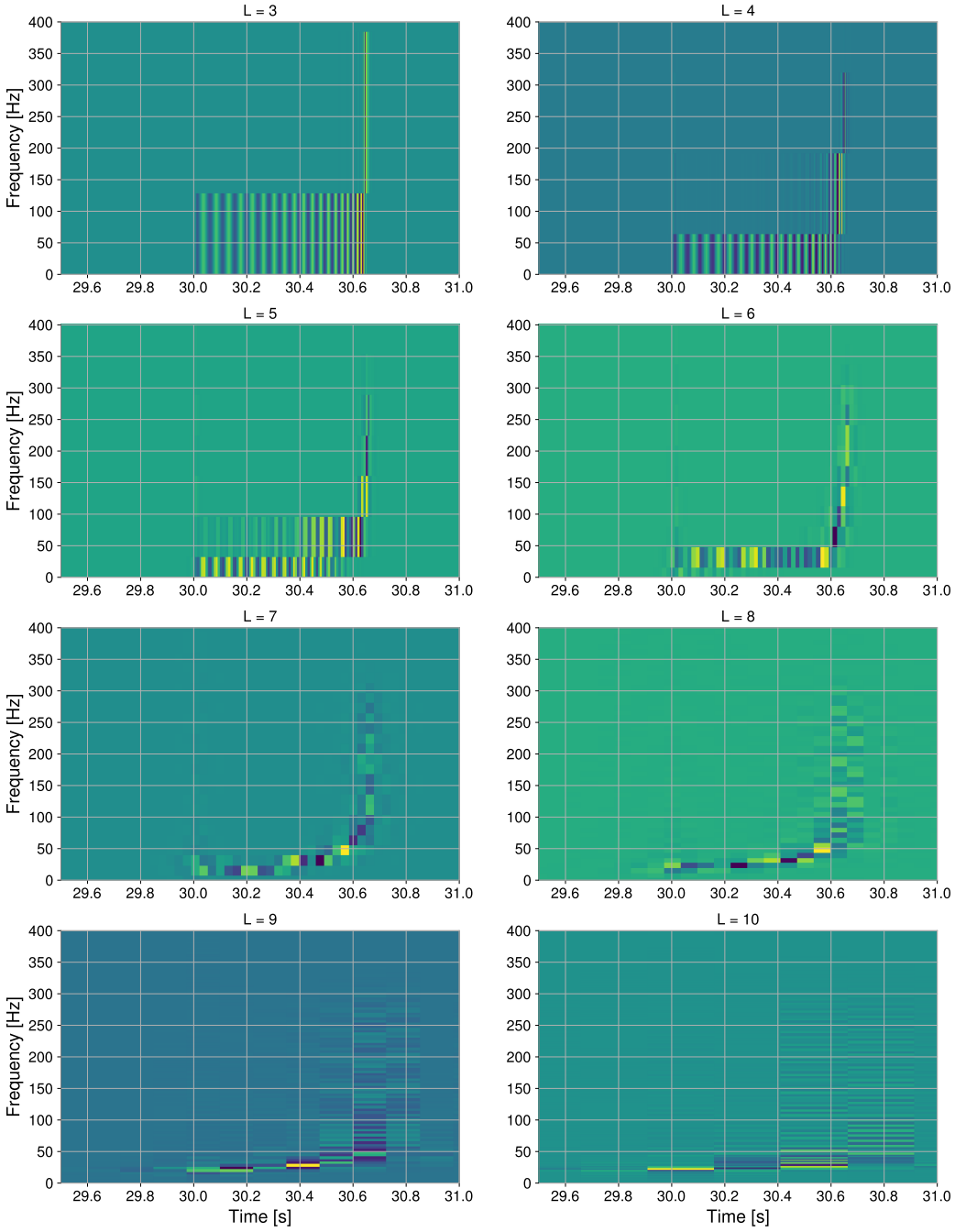


Figure 3.2: The result of the WDM transform of a zero-noise CBC signal of equal masses $M_1 = M_2 = 40M_\odot$ at a sampling frequency of 4096 Hz and different resolution levels from L=3 to L=10

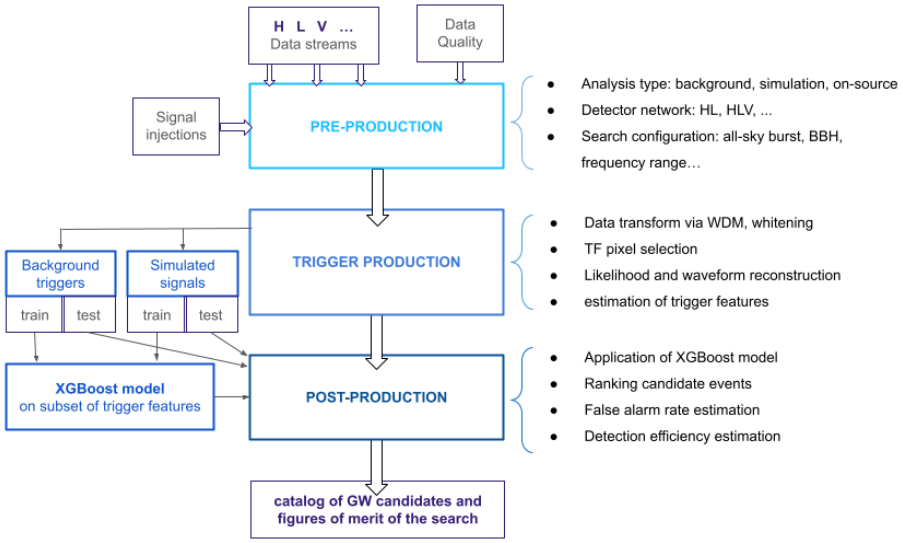


Figure 3.3: The cWB workflow. The analysis starts with data conditioning, which includes regression and whitening. After data conditioning, the pixel selection algorithm identifies clusters of pixels with excess power in the time-frequency plane. Finally, coherent statistics are computed to distinguish between noise and signal. Figure from [87]

3.2 Coherent WaveBurst

Having introduced the WDM transform, we can now present cWB 2G and its workflow. cWB [71, 56] is a pipeline designed to search for GWs using a minimal set of assumptions, without targeting any specific signal morphology or relying on a specific waveform model. In cWB, transients are represented in the time-frequency plane as localized excess of energy with respect to the floor defined by the energy of the noise. Since the noise is not gaussian but displays non-gaussian noise transients that can mimic the presence of signal in the data, it is fundamental to find a way to distinguish the origin of such an excess. To overcome this problem, cWB leverages the concept of "coherence" to distinguish between noise transients and genuine signals. The idea is the following: noise transients are "local" excess of energy at the photodetector and are completely uncorrelated among different detectors[1]. GWs instead are expected to be detected by different detectors in the network, with a time delay consistent with the time of flight between the detectors.
This means that, local excess of energy that are only seen by one detector are most likely of instrumental origin, while genuine candidate should be found in multiple detector's outputs. This idea is behind the concepts of "coincide" and "coherence" [24]. Coincidence searches firstly look for an excess of energy in a single detector and tries to find a similar excess

[1]The assumption of uncorrelated noises in distant detectors passed all empirical tests to date in the context of gravitational wave transient searches on data compliant with science quality. In particular, for what concerns our data analysis methods, in [87] we tested that the statistical properties of the coherent trigger candidates showing up accidentally does not depend on the time shifts implemented between detectors' data streams. This empirical evidence, though, cannot fully rule out sources of correlation travelling at light speed, which would appear compatible with arrival times of genuine GW transients and in principle could mimic a GW.

in the other detectors within a time window consistent with the time of flight between the detectors. Coherence searches instead incorporate this idea in the whole search, with proper combination of the data from different detectors, so that the search is performed on the combined data from the network of detectors, rather than on single-detector data. cWB is implemented to perform coherent searches. For this reason, cWB can only work with detectors network and not with single detectors.

Another important feature of cWB is its multi-resolution approach. Gravitational wave signals can exhibit a wide range of time-frequency morphologies, from short-duration bursts to long-duration signals with complex frequency evolution. To accommodate this diversity, cWB performs the analysis at multiple resolution levels l allowing it to capture signals with varying time-frequency characteristics effectively. The resolution levels are also chosen by the user and they range from a minimum defined as l_{low} and a maximum called l_{high}.
Going from one resolution level i to the "following" one $i+1$ corresponds to a scaling of the time and frequency resolution by a factor of two, i.e. $\sigma_t^{i+1} = 2\sigma_t^i$ and $\sigma_f^{i+1} = \frac{1}{2}\sigma_f^i$.

Having set up the analysis, cWB can be run to analyse the data for the chosen network of detectors.

Figure 3.3 illustrates the complete steps of the cWB workflow, from the configuration of the analysis to the production of the candidate signals with their relative statistical significance. In summary, the cWB analysis workflow is organized into four main stages.

i) the **pre-production** phase defines the configuration of the search, including the detector network, frequency band, time-frequency resolution, analyzed livetime, possible sky constraints, and the management of computing jobs. This stage can be adapted to different analysis modes, such as on-source searches, background estimation through time shifts, and simulation studies with injected signals.

ii) The **production of candidate triggers** and their summary statistics. It begins with data pre-conditioning, where spectral lines are reduced through regression and the data are whitened. Trigger candidates are then selected in the time-frequency plane by identifying coincident excess power across detectors, and the surviving candidates are processed through a coherent likelihood maximization to estimate their physical properties and relevant features.

iii) The **post-production** stage involves ranking the candidate triggers using an XGBoost classifier (a supervised learning algorithm based on gradient-boosted decision trees) trained on background and simulated events, allowing the pipeline to distinguish accidental noise triggers from true signal-like candidates.

iv) the results are reported as a catalog of candidate GW transients, together with their reconstructed properties.

3.2.1 Data Conditioning

The first step of the analysis is **data conditioning**. Data conditioning is necessary to remove correlations within the data, so that the autocorrelation matrix R_{ij} of the data becomes the identity matrix:

$$R_{ij} \propto \delta_{ij}, \tag{3.15}$$

transforming the data into a white noise process. In the time domain, this means that the data are uncorrelated in time, while in the frequency domain, it means that the PSD of the data is flat across the entire frequency band. In the Fourier domain, if the data were stationary and infinite dimensional, this could be achieved by dividing the data by the square root of the PSD $S_i(f)$:

$$\omega_i[f] = \frac{x_i[f]}{\sqrt{S_i(f)}},$$

such a transformation is defined as "whitening" of the data. However, since we can only deal with a finite number of samples in the data, for which stationarity can only be assumed locally, steps other than whitening are necessary to pre-process the data. Therefore, before whitening the data, cWB performs another step called "regression", which removes sharp spectral lines in the PSD.

Regression

For GW interferometers, achieving a flat PSD is difficult because the noise spectrum exhibits both smooth broadband variations and sharp spectral peaks at specific frequencies. To remove these spectral lines, cWB applies a time-frequency regression procedure [113] that models the coupling between the target GW strain channel (h) and auxiliary environmental witness channels ($x, y, \dots$). Within each wavelet layer, the regression algorithm computes finite impulse response filter coefficients ($a_j, b_k, \dots$) of length $2L + 1$ by minimizing the mean-square error, χ^2, between the target data and the predicted noise:

$$\chi^2 = \sum_i \left(h[i] - \sum_{j=-L}^{L} a_j x_{i+j} - \sum_{k=-L}^{L} b_k y_{i+k} - \dots \right)^2 . \tag{3.16}$$

Once the system is solved, the optimal coefficients are applied back to the auxiliary channels to synthesize the predicted environmental noise signature, $n[i]$:

$$n[i] = \sum_{j=-L}^{L} a_j x_{i+j} + \sum_{k=-L}^{L} b_k y_{i+k} + \dots \tag{3.17}$$

This synthesized noise is then subtracted directly from the target GW channel ($h[i] - n[i]$), effectively mitigating sharp peaks in the PSD and making the data more suitable for subsequent whitening. Figure 3.4 illustrates the effect of this regression on the PSD. As shown in the figure, regression successfully removes most sharp peaks in the PSD. However, for complex structures like those around 500 Hz, the algorithm significantly reduces their magnitude but does not completely eliminate them. Following this regression step, the cleaned data are ready for the final whitening stage.

Whitening

After the regression, the final step in data conditioning is the whitening. In cWB, whitening is performed in the time frequency domain.
The goal of whitening is to transform the data into a white noise process, where the PSD is flat, and each sample is independent in both time and frequency domains. Dividing by the PSD means that each frequency component is normalized to have unit variance. The standard whitening procedure in cWB consists of two main steps:

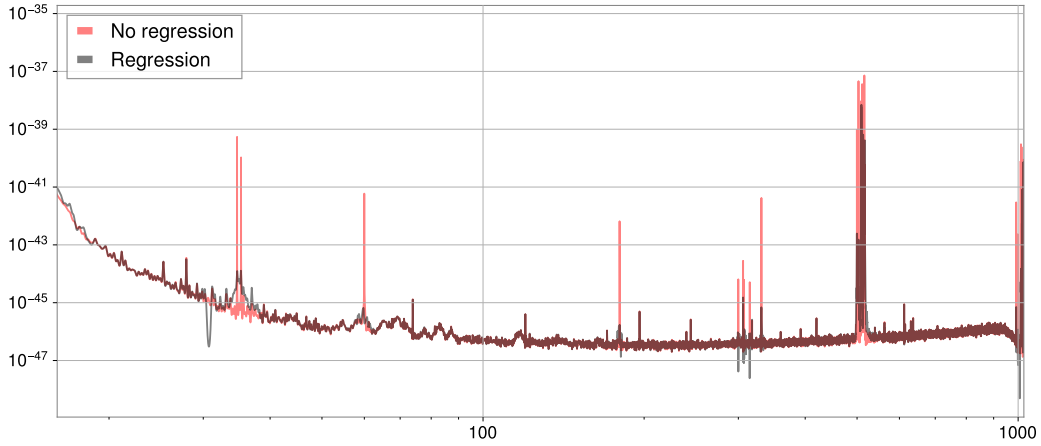


Figure 3.4: Effect of regression on the PSD. The red solid line represents the PSD before regression, while the black solid line shows the PSD after regression. Regression effectively removes sharp peaks, though some complex structures, such as those near $f \sim 500$ Hz, are only partially mitigated.

- Computation of the noise root-mean-square (**nRMS**) in the time-frequency domain.
- Whitening the data by dividing the WDM coefficients by the corresponding nRMS values.

The computation of the nRMS is the core of the whitening, since they serve as the normalization factor for the whitening process. Instead of relying to a spectral estimator to compute the PSD, cWB implement an estimator of the noise energy that is robust to the presence of strong outliers, may them be signals or glitches.
To compute the nRMS, the time-frequency data are first divided into a sequence of overlapping windows of duration 60 seconds. The windows are placed with a stride of 20 seconds so that consecutive windows start at times t_0, $t_0 + 20\,\mathrm{s}$, $t_0 + 40\,\mathrm{s}$, and so on. Each window, therefore, overlaps the previous and the next one by 40 seconds. Given a frequency window, for every frequency layer of the WDM, the nRMS is computed as a "robust" estimate of the standard deviation over all the time bins. Instead of using the standard deviation, which is sensitive to outliers and can be be affected by the presence of non gaussian noise, the standard deviation is computed as the difference between the 84.15 and the 15.87 percentile, whose distance (under the normality) assumption, is exactly equal to 2σ. This defined the energy of the noise process as:

$$\mathrm{nRMS} = \frac{P_{84.15\%} - P_{15.87\%}}{2}. \tag{3.18}$$

together with the nRMS, the median M over the pixel is also computed and removed to the data. With the above procedure, series of estimates of the nRMS and the median M are obtained for every frequency bin, but separated by 20 seconds in time. To cover all the time bins, the nRMS and the median are linearly interpolated over time, so that every pixel in the time-frequency plane has a corresponding nRMS and median value, and the whitened time-frequency map is obtained by subtracting the median and dividing by the nRMS, and

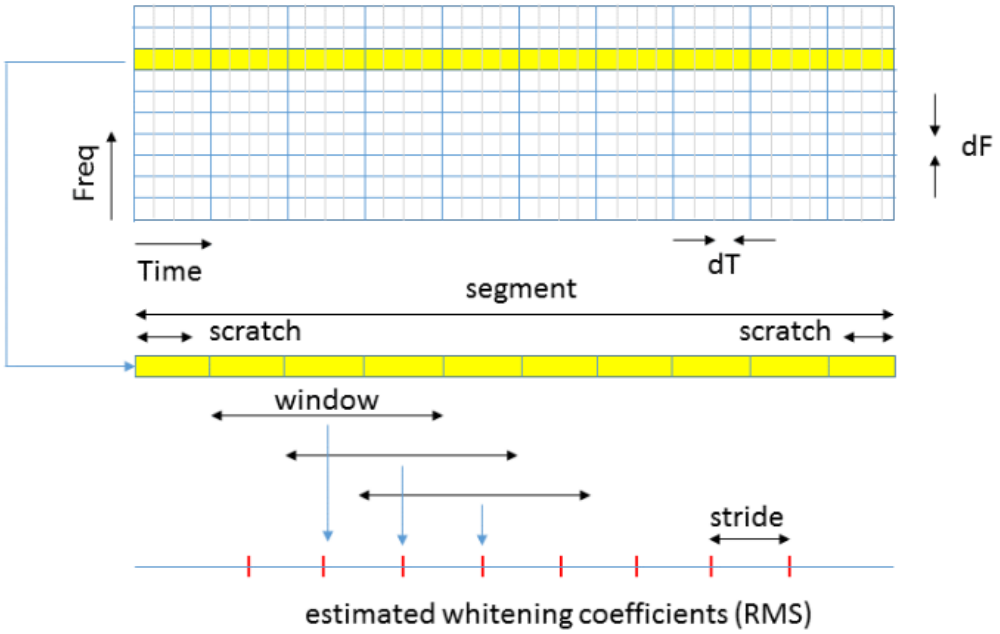


Figure 3.5: The WDM whitening procedure. nRMS values are computed in the time-frequency domain and used to normalize the WDM coefficients, transforming the data into a white noise process. Figure adapted from Coherent WaveBurst documentation [37].

the final whitened WDM coefficients are given by:

$$\omega(t,f)_W = \frac{\omega(t,f) - \mathrm{M}(t,f)}{\mathrm{nRMS}(t,f)}. \tag{3.19}$$

Once the data are whitened, we obtain a time series whose noise has an approximately flat PSD[2] with zero mean, and unit variance. In this whitened state, the time-frequency representation should not exhibit any discernible patterns, with localized energy excesses being sparse and lacking structure in the time-frequency plane. The next step is to identify whether there are localized excesses of power in both time and frequency, which could indicate the presence of a signal in the data. The cWB software employs a **pixel selection** algorithm to detect excess power in the data. At the different resolution levels, only a small fraction of pixels are selected. The selection is performed on the pixel whose energy is large than a certain fixed threshold, that depends on the probability that a certain pixel is not produced by noise. The probability is approximately equal to the fraction of loud pixels selected by the algorithm for each TF resolution. The pixels above the thresholds are then selected together with their neighboring pixels, according to some pattern that can be set by the user (for example, for chirps, the pixel can be selected to follow a diagonal pattern in the time-frequency plane), and the connected, selected pixels are grouped together to form a cluster. Each cluster can be composed by pixels belonging to different time-frequency resolutions, and is the basic unit for the next step of the analysis, where the coherent statistics are computed to distinguish between noise and signal.

[2]Spectral flatness is inherently limited by the frequency resolution of the WDM, as finer spectral structures remain unresolved by the transform.

3.2.2 Coherent Statistics

The clusters saved by the previous steps have to be analyzed to distinguish between noise and signal. As mentioned in the previous section, cWB combines the data from different detectors to compute coherent statistics and assess the likelihood of the presence of a signal in the data.
When analyzing data from a network of detectors, cWB computes a set of coherent statistics to test two competing hypotheses:

- H_0: The data are consistent with the noise hypothesis, i.e., the data represent a noise process.
- H_1: The data are consistent with the signal hypothesis, i.e., the data represent a noise process plus a signal.

Under the H_0 hypothesis, the data can be expressed as $x[t] = n[t]$, where $n[t]$ represents pure noise. Under the alternative hypothesis H_1, the data are modeled as the sum of detector noise $n[t]$ and the detector response to the signal $\xi[t]$, i.e.,

$$x[t] = n[t] + \xi[t].$$

The cWB software computes coherent statistics to test these hypotheses. The likelihoods under the two hypotheses are defined as:

$$\mathcal{L}_0 = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_i^2}{2\sigma^2}\right), \tag{3.20}$$

$$\mathcal{L}_1 = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - \xi_i)^2}{2\sigma^2}\right). \tag{3.21}$$

To simplify computations, it is convenient to work with complex waveforms u and complex antenna patterns A, defined as:

$$u = h_+ + \imath h_\times, \tag{3.22}$$

$$A = F_+ + \imath F_\times. \tag{3.23}$$

Using these definitions, the detector response can be expressed as:

$$\xi = A^* u + A u^*,$$

where the asterisk denotes complex conjugation. A likelihood ratio test can then be constructed to compare the two hypotheses:

$$\mathcal{L} = \log\left(\frac{\mathcal{L}_1}{\mathcal{L}_0}\right) = \sum_{i=1}^{N} \left(\frac{x[i]\xi[i]}{\sigma^2} - \frac{\xi[i]^2}{2\sigma^2}\right).$$

For a network of K detectors, the likelihood is the sum of the single-detector likelihoods:

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{i=1}^{N} \left(\frac{x_k[i]\xi_k[i]}{\sigma_k^2} - \frac{\xi_k[i]^2}{2\sigma_k^2}\right).$$

Rewriting the detector response in terms of the complex antenna pattern and waveform, the likelihood becomes:

$$\mathcal{L} = \sum_{i=1}^{N} \left(u[i]X^*[i] + u^*[i]X[i] - g_r u[i]u^*[i] - \frac{g_c^* u[i]^2 + g_c u^*[i]^2}{2} \right).$$

Here, the network variables X, g_c, and g_r are defined as:

$$X[i] = \sum_{k=1}^{K} \frac{A_k x_k[i]}{\sigma_k^2}, \tag{3.24}$$

$$g_c = \sum_{k=1}^{K} \frac{A_k A_k}{\sigma_k^2}, \tag{3.25}$$

$$g_r = \sum_{k=1}^{K} \frac{A_k^* A_k}{\sigma_k^2}. \tag{3.26}$$

These definitions allow the likelihood to be expressed in terms of network variables rather than single-detector variables. For example, $X[t]$ represents the network output time series, which is a weighted combination of single-detector time series. Both $X[t]$ and g_c are complex numbers by definition, while g_r is the only real number among them. These variables allow the likelihood for a network of detectors to be described by "marginalizing" over single-detector variables, providing a new representation of GW interferometers as a network. To simplify further, the system can be rotated into the **Dominant Polarization Frame** (DPF)[74, 71], where g_c becomes a real number. Since g_c is a complex number, it can always be written as $g_c = |g_c|e^{\imath\gamma}$ for some phase γ, so a rotation removing this phase can always be defined:

$$A' = Ae^{\imath\gamma/2}, \quad u' = ue^{\imath\gamma/2}, \quad g_c' = g_c e^{-\imath\gamma}. \tag{3.27}$$

This rotation always exists, for any network of detectors, because it is equivalent to a singular value decomposition [109] of the matrix $\boldsymbol{\mathcal{F}}$ built from the antenna patterns of every detector defined as

$$\boldsymbol{\mathcal{F}} = \left(\mathbf{F}_k^+, \mathbf{F}_k^\times\right) \tag{3.28}$$

with ranging over the detectors of the network. Removing the phase γ from g_c is the same operation as diagonalizing that matrix, so the DPF can always be reached regardless of the network configuration. In this representation, the detector antenna patterns become:

$$\mathbf{f}_1 = \mathbf{f}_+ \cos(\gamma) + \mathbf{f}_\times \sin(\gamma) \tag{3.29}$$

$$\mathbf{f}_2 = -\mathbf{f}_+ \sin(\gamma) + \mathbf{f}_\times \cos(\gamma) \tag{3.30}$$

This ensures that the Network Antenna Pattern for the $\mathbf{f}_1$ and $\mathbf{f}_2$ become orthogonal in this representation. In the DPF, the network response is expressed as:

$$R(u) = gh_1 + \epsilon g h_2,$$

where $g = g_r + |g_c|$ and $\epsilon = \frac{g_r - |g_c|}{g}$. The parameter ϵ (network alignment factor) ensures $0 \leq \epsilon \leq 1$, guaranteeing that $|\boldsymbol{f_2}| \leq |\boldsymbol{f_1}|$, which explains the name Dominant Polarization Frame. The DPF is a particularly useful representation for coherent analysis, as it simplifies the likelihood expression and facilitates the computation of coherent statistics. The above

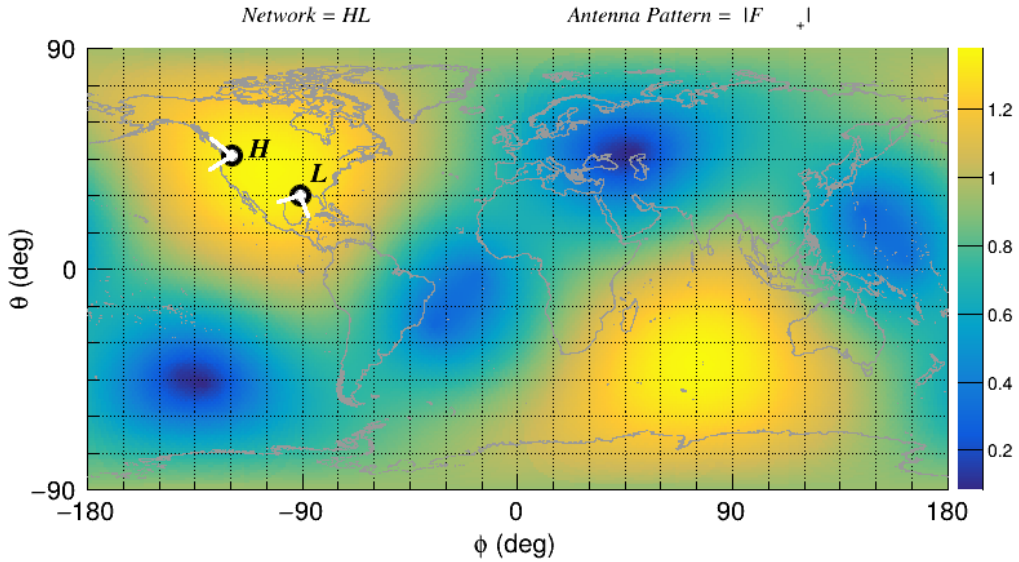


Figure 3.6: The Network Antenna Pattern for the F_1 in the Dominant Polarization frame for LIGO-Livingston (L) and LIGO-Hanford (H) detectors Network (LH).

discussion was presented in the time domain for the whole dataset, but it can be naturally extended to the time-frequency domain, in which cWB analyzes clusters of pixels $\mathcal{C}$, where $i \in \mathcal{C}$. In the time frequency domain, the network response and noise-scaled data are defined as:

$$\boldsymbol{\xi}[i] = \boldsymbol{\mathcal{F}}[i]\boldsymbol{h}[i], \quad \boldsymbol{w}[i] = \left(\frac{x_1[i]}{\mathrm{nRMS}_1[i]} \quad \cdots \quad \frac{x_K[i]}{\mathrm{nRMS}_K[i]} \right)^T .$$

Here $\boldsymbol{h}[i] = (h_1[i], h_2[i])^T$ is the vector of the two waveform polarizations in the DPF, $\boldsymbol{w}[i]$ is the K vector of whitened data defined above, and $\boldsymbol{\mathcal{F}}[i]$ is the $K \times 2$ matrix collecting the (noise-weighted) antenna patterns of the K network detectors for the two DPF polarizations. We denote its two columns by $\boldsymbol{f_1}[i]$ and $\boldsymbol{f_2}[i]$, i.e. $\boldsymbol{\mathcal{F}}[i] = (\boldsymbol{f_1}[i] \;\; \boldsymbol{f_2}[i])$, so that $\boldsymbol{\xi}[i]$, the whitened network response, is the K vector resulting from their product with $\boldsymbol{h}[i]$.

The likelihood ratio can then be rewritten as:

$$\mathcal{L} = 2\,\langle \boldsymbol{w}|\boldsymbol{\xi}\rangle - \langle \boldsymbol{\xi}|\boldsymbol{\xi}\rangle\,,$$

where, for two network vectors $\boldsymbol{a}[i]$ and $\boldsymbol{b}[i]$, we define the inner product $\langle \boldsymbol{a}|\boldsymbol{b}\rangle \equiv \sum_{i=1}^{N} \mathbf{a}[i]\mathbf{b}[i]$ as the standard product over the K detectors of the network, and the sum is performed over all pixels in the cluster. By maximizing the likelihood ratio with respect to the two polarizations h_1 and h_2, we obtain a system of equations for the two waveform polarizations in the DPF, which are:

$$h_1[i] = \frac{\boldsymbol{w}[i] \cdot \boldsymbol{e_1}[i]}{|\boldsymbol{f_1}[i]|}, \tag{3.31}$$

$$h_2[i] = \frac{\boldsymbol{w}[i] \cdot \boldsymbol{e_2}[i]}{|\boldsymbol{f_2}[i]|}, \tag{3.32}$$

where $\boldsymbol{e_1}[i]$ and $\boldsymbol{e_2}[i]$ are unit vectors along the DPF antenna patterns, defined as:

$$\boldsymbol{e_1}[i] = \frac{\boldsymbol{f_1}[i]}{|\boldsymbol{f_1}[i]|}, \quad \boldsymbol{e_2}[i] = \frac{\boldsymbol{f_2}[i]}{|\boldsymbol{f_2}[i]|}.$$

The likelihood ratio statistics is obtained by substituting the solutions for h_1 and h_2 back into the likelihood ratio expression. Substituting these solutions into the likelihood ratio, the statistic is given by:

$$\lambda_{LR} = \sum_i \boldsymbol{w}[i] P[i] \boldsymbol{w}^T[i], \tag{3.33}$$

where P is the projection operator onto the network plane:

$$P_{nm}[i] = \boldsymbol{e_1}_n[i] \boldsymbol{e_1}_m[i] + \boldsymbol{e_2}_n[i] \boldsymbol{e_2}_m[i]. \tag{3.34}$$

The likelihood ratio statistics, being a quadratic form, can be interpreted as the total energy of the network output in the whitened domain, i.e. is also proportional to the signal to noise ratio of the network. If the data were perfectly gaussian, the discussion developed so far would be sufficient to distinguish between signals and noise. In fact, the distribution for the signal to noise ratio is known in such a case and the likelihood ratio statistics would be sufficient to perform the hypothesis test. The noise in the detectors is not purely gaussian, which means the distribution for the likelihood ratio statistics is not known, and under the null hypothesis is not easily recovered. Even worse, there are noise triggers in the data which can produce a high value of the likelihood ratio statistics as well. One more representative statistics come from the re-writing of the likelihood ratio statistic 3.33 in terms of two different components, called coherent and incoherent energy:

$$E_c = \sum_i \sum_{n \neq m} w_n[i] P_{nm}[i] w_m[i], \tag{3.35}$$

$$E_i = \sum_i \sum_{n = m} w_n[i] P_{nm}[i] w_m[i]. \tag{3.36}$$

E_i is called the incoherent energy, and is in-fact built buy the sum of the diagonal elements of the projection operator, and therefore it describes the contribution of a "single detector" to the total energy of the network. E_c is called the coherent energy, and is built by the sum of the off-diagonal elements of the projection operator, and therefore describes the contribution of the "cross terms" to the total energy of the network, i.e. it takes into account the contribute of different pairs of detectors.

From these quantities, one can build network correlation coefficient cc, defined as:

$$cc = \frac{E_c}{E_c + E_i},$$

which measures the fraction of coherent energy relative to the total energy of the network output.

3.3 Candidate Selection and significate

So far, we have shown how the pipeline loads the data, preprocesses them, selects potential triggers, and computes the likelihood ratio statistics in terms of coherent and incoherent energy. Anyway in case of real data we have to considered not only the gaussian noise but

also the glitches. While coherence is a powerful statistic to distinguish gravitational signals from noise, some glitches can also exhibit a degree of coherence between the detectors, hence cannot be used as an optimal detection statistics. The detection statistics ρ_0 [74] for cWB was

$$\rho_0 = \sqrt{\frac{E_c}{1 + \chi^2 \cdot (\max(1, \chi^2) - 1)}} \tag{3.37}$$

where $\chi^2 \equiv E_n/N_{df}$ is a proxy of the Chi-square statistics [37] and N_{df} is the number of independent wavelet amplitudes used to characterize the trigger. ρ_0 has been the detection statistic for several years, but has been recently modified by introducing an XGBoost classifier [35] (a short description is provided below) to discriminate between genuine signals and noise transients. After the XGBoost introduction, the detection statistic is now the XGBoost score W_{XGB}, trained to distinguish between signals and glitches [110, 87]:

$$\rho = \mathrm{W}_{XGB}. \tag{3.38}$$

When referring to the detection statistic in the following, we refer to the latest definition, while the previous ones will be referred to as ρ_0.
The XGBoost classifier is a deep forest model used to perform supervised learning on tabular data. For this reason, it needs to be trained on a set of features that it learns to associate with the two classes: glitches [0 class] and signals [1 class]. A comprehensive list of these features can be found in [87], and they are produced by cWB for each and every candidate trigger. Most of these features are designed to be agnostic to the morphology of the candidate signal since the aim is to be sensitive to the widest range of candidate signals. Only two features (namely Q_a and Q_p) are related to morphology and have been created to suppress a specific class of glitches called blips [32, 17]. The definitions of Q_a and Q_p are given in [29], along with an alternative method to suppress blip glitches using autoencoders.
To create instances of the signal class (label = 1) for the XGBoost supervised learning, signal simulations are performed and injected into the real data. To ensure generality, the training injections are carried out using White Noise Bursts (WNB) [73]. WNB signals are generated by restricting white noise to specific time-frequency regions. Their durations and frequency spans cover a broad range of values. Two sets of simulations are considered:

(i) WNBs with central frequencies uniformly distributed in the range $[24, 996]$ Hz, bandwidths between $[10, 300]$ Hz, and durations logarithmically distributed between 0.1 ms and 1 ms.

(ii) WNBs with central frequencies logarithmically distributed in the range $[24, 450]$ Hz, bandwidths between $[10, 250]$ Hz, and durations between 1 ms and 50 ms.

The choice of WNBs is due to the generality of Burst searches. WNBs can cover a wide range of possible signal morphologies and are not biased toward any specific model. In this way, it is ensured that the XGBoost is not trained on any specific morphology.

The way instances of the noise class are built is more complex and is obtained via the "time-slides" method.

3.3.1 Building Instances of the Noise Class

To assess the significance of each candidate, it is essential to construct a set of noise triggers that represent instances of the noise class. This is achieved using the "time-slides" procedure, a type of analysis referred to as "background" analysis (BKG). The time-slides procedure

is fundamental for estimating the False Alarm Rate (FAR) or its inverse, the iFAR, which quantifies the statistical significance of each candidate.

The time-slides procedure involves shifting the time series of one or more detectors relative to the others by a time delay larger than the maximum time delay a GW signal traveling at c can have within the network. For example, in the LIGO Hanford and LIGO Livingston network, the maximum time delay between detectors is approximately $10ms$. By shifting the time series of one detector by any longer time delay, it is guaranteed that no genuine signal can be present in the time-shifted data. Consequently, any trigger identified in the time-shifted data can be safely assumed to be an instance of the noise class. For the O4 analysis the time shift is set to be 0.5 seconds.

The time-shifted data are analyzed using the same workflow as the real data, and the triggers identified in the time-shifted data are treated as noise triggers. This procedure can be repeated multiple times, applying different time delays to increase the statistical sample of noise triggers. The total duration of the time-shifted data analyzed is referred to as the "background livetime," which is typically much larger than the real livetime of the analysis. The background livetime is then used to compute the FAR or IFAR for each candidate, providing a measure of its statistical significance.

3.4 py-Coherent WaveBurst

The previous discussion focused on the workflow of the Coherent WaveBurst 2G software, whose revised and production version is written in C++ and ROOT. However, the core algorithms of cWB have been re-written in a Python package called pyCoherent WaveBurst (pyCWB) [123, 122].
Despite being at the forefront of GW research and one of the most sensitive pipelines designed The core code is complex and not easy to modify, making it challenging to implement new features or test new ideas. Several other GW pipelines are now written in Python, such as PyCBC [93], Bilby [26], among others. Transitioning to Python and leveraging the modularization of Python packages can enable the creation of more user-friendly and flexible software, which can be easily modified and adapted to new requirements.

Figure 3.7 shows a comparison of the execution time distributions for the pyCWB and cWB packages on the chunk 23 (K23) of O4, the total execution time for pyCWB is roughly double the execution time of cWB. The pyCWB package is designed to closely replicate the C++ version of cWB, maintaining the same workflow and algorithms. Most of the code is a wrapping of the C++ version achieved via the pyROOT package, but new features can be added using pure Python modules. The pyCWB package is still under development, and some results and enhancements from this PhD thesis work have been implemented in pycwb. While the pyCWB package is yet used for official LVK searches, it is possible to perform complete analysis with it. The results presented in this theses have all been obtained using pyCWB, and the methodological advancements are now part of the pyCWB package.

3.4.1 The pyCWB structure

The pycWB package is structured into several modules, each of them is responsible of a specific task for the analysis

Figure 3.8 shows the structure of the pyCWB package. The first two folders, the "config" and "constant", are used for the "pre-analysis" stage, to configure the type of analysis (Background analysis, Injection analysis or Real analysis) and to define the configuration of

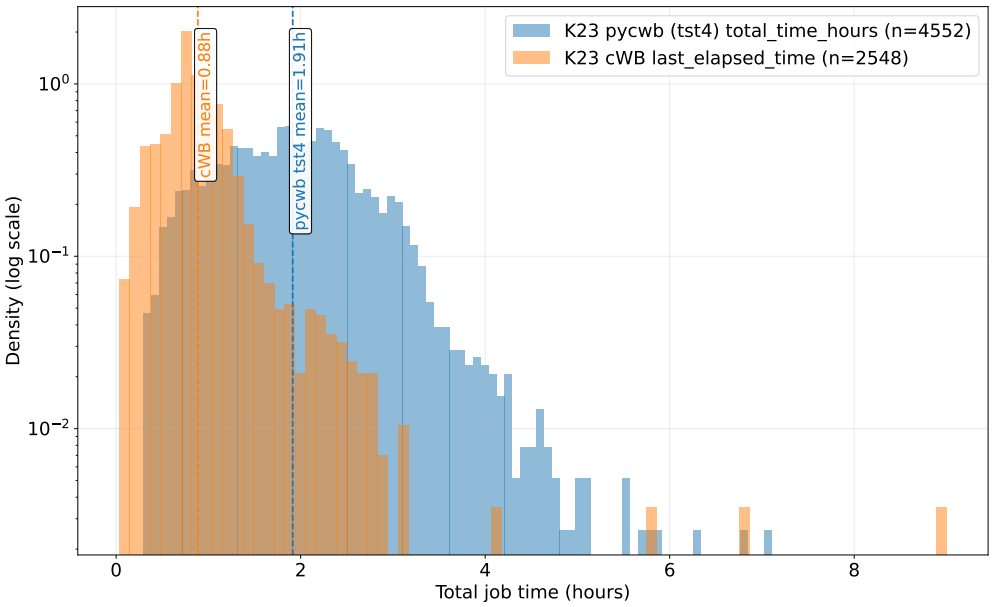


Figure 3.7: Comparison of the execution time distributions for the pyCWB and cWB packages on the chunk 23 (K23) of O4. The execution time form pycWB is roughly double the execution time of cWB. Figure courtesy of Y. Xu

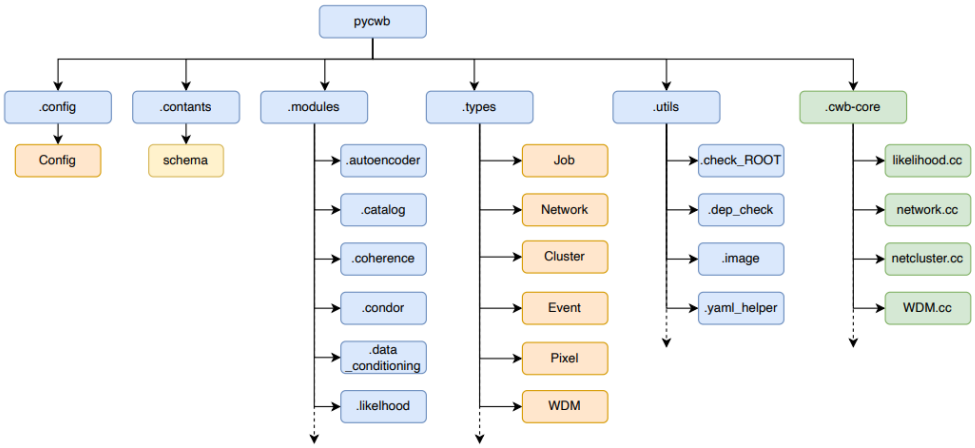


Figure 3.8: Figure from [123]. The structure of the pyCWB package. The package is structured into several modules, each responsible for a specific task in the analysis. The core algorithms are wrapped from the C++ version of cWB using the pyROOT package, while new features can be implemented using pure Python modules. The green moudles represent the core cWB code, transposed here to be wrapped with pyROOT in pycWB. The orange folders contain all the needed types to perform the analysis, while the blue folders represent the collection of all the algorithms needed to run a full analysis workflow. This scheme is rapidly evolving and a completely pythonized version is now available.

the analysis, such as the the Network of detectors (HL, HLV, ...), the type of injections to be used and so on. The "constant" fonder contains the parameters schema, i.e. the list of all the parameters that can be used to configure the analysis, together with their default values. The config file is implemented as a yaml file, which is a human-readable data serialization standard that can be used in conjunction with all programming languages and is often used to write configuration files.
The analysis modules are divided into the three main folder: "cwb core", "types" and "modules". The "cwb core" folder contains the core C++ algorithms of the original cWB pipeline, with its types and methods. The "types" folder contains the data structures used throughout the code, such as the detectors, the triggers, the cluster and so on, with their specific methods. The modules folder contains the main modules of the analysis, each of them is responsible for a specific task. There are several modules that take care of the data conditioning, using the regression and the whitening, the coherence, the likelihood computation, and so on.

All of the above folders and modules are used to perform the analysis, which is implemented via some specific workflow, that is the final folder of the package and combines all the modules in a simple and readable way that can be modified and adapted. Creating a new workflow is easy and allow to implement different types of analysis to meet specific needs.

The last step, which is the post production, gains a lot from the Python implementation, since several Python packages are available to create plots, tables and so on. Also specific packages for machine learning can be used, such as pytorch, sklearn, XGBoost and more. Thanks to the modularization, it is easy to implement and test new post production techniques, with a simple addition of a new module and lines of codes.

Chapter 4

Maximum Entropy Spectral Analysis

4.1 Maximum Entropy Spectral Analysis

The aim of this chapter is to introduce the Maximum Entropy Spectral Analysis (MESA) method for the estimation of PSDs from time series data. MESA is a powerful technique that leverages the principle of maximum entropy [106] to provide high-resolution spectral estimates, particularly useful when dealing with short or noisy data sets.
In here we focus on MESA method as an approach to estimate PSD, without going into the details of the Maximum Entropy principle itself. For a more complete introduction to the subject, the reader is referred to [88].
On its first definition, by Shannon [106], the entropy of a probability distribution $p(x)$ is defined as

$$H(p) = -\int p(x) \log p(x) dx. \tag{4.1}$$

It means that, an event with probability p has got an information content of $-\log p$. The entropy is then the average information content of the distribution, which means that the higher the entropy, the less information we have on the system described by the probability distribution.
The Maximum Entropy Principle [68, 69] states that, given a set of constraints on the probability distribution, the distribution that maximizes the entropy is the one that best represents our knowledge of the system, without introducing any spurious information, i.e. it is the least biased.
This principle can be reformulated in the context of the estimation of PSDs for time series data. In the context of GWs, as we showed in Chapter 2, the knowledge of the PSD is crucial for the analysis, therefore building good spectral estimators is of paramount importance.
This is not only true in the context of CBC analysis, but also for burst searches, where the data are whitened before the analysis. The Maximum Entropy Principle, as shown by Burg [31], can be exploited to develop a completely new way for the estimation of PSDs, with only few requirements: i) the PSD estimate has to be non-negative; ii) its Fourier transform has to match the sample autocorrelation.
To do so, there is a problem we have to solve: our definition of entropy in Equation 4.1 depends on the probability distribution, not the PSD Therefore, the variational problem

must be re-written in terms of the PSD $S(f)$ alone. A way out of this problem is considering our signal as the result of filtering a white noise process with a filter whose power response is exactly $S(f)$ [15]. One can show that the 'entropy gain' obtained filtering white noise is:

$$\Delta H = \int_{-\infty}^{\infty} \log S(f) df, \tag{4.2}$$

so that we can reformulate the variational problem requiring $S(f)$ to maximize the entropy gain. In real examples we have no access to all frequencies but only on those between minus and plus the Nyquist frequency, defined from the sampling interval Δt as

$$N_y = \frac{1}{2\Delta t}. \tag{4.3}$$

All our quantities are well defined in this range only, so that we are going to maximize

$$\Delta H = \int_{-N_y}^{N_y} \log S(f) df. \tag{4.4}$$

Defining $\bar{r}_n$ as the sample autocorrelation, the constraint is:

$$\int_{-N_y}^{N_y} S(f) e^{\imath 2\pi f n \Delta t} df = \bar{r}_n. \tag{4.5}$$

This approach on PSD estimate was developed by Burg [31] and it is a way out from the problems raised in periodograms estimate. It requires no assumptions on the unavailable estimates for $\bar{r}_n$ and is consistent with the sample autocorrelation function by construction. The hard task is now to show that this problem has a solution.

4.2 MESA solution

We follow the solution proposed by Burg in his PhD thesis [31]. Instead of directly solving the variational problem for the PSD, we rewrite Equation 4.4 by constraining $S(f)$ to be the Fourier Transform of the sample autocorrelation function $\bar{r}_n$, as:

$$S(f) = \frac{1}{2N_y} \sum_{n=-\infty}^{\infty} \bar{r}_n e^{-\imath 2\pi n f \Delta t}. \tag{4.6}$$

Substituting in 4.4, the variational problem reduces to the maximization of:

$$\int_{-N_y}^{N_y} \log \left(\frac{1}{2N_y} \sum_{n=-\infty}^{\infty} \bar{r}_n e^{-\imath 2\pi f n \Delta t} df \right). \tag{4.7}$$

As we said before, not all the values for the autocorrelation are known. Generally, we have access to a finite number of values for the autocorrelation function, ranging in a symmetric interval $n \in [-N, N]$. In the previous equation, since the summation index is ranging between $-\infty$ and $+\infty$, we are correctly assuming that also the unknown values have to be used in the computation of the PSD. But, since they are not known, we have to require the maximization of entropy to be independent of these values:

$$\frac{\delta H}{\delta \bar{r}_s} = 0 \text{ for } |s| > N.$$

Using the explicit form for H, this becomes

$$\frac{\delta H}{\delta \bar{r}_s} = \frac{1}{2N_y}\int_{-N_y}^{N_y} S(f)^{-1} e^{-\imath 2\pi f s \Delta t} df = \lambda_s = 0 \text{ for } |s| > N. \tag{4.8}$$

We see that $S(f)$ can be expressed in terms of the λs via a Fourier Series

$$S(f)^{-1} = \sum_{s=-N}^{N} \lambda_s e^{-\imath 2\pi f s \Delta t}, \tag{4.9}$$

This is a non-linear closed expression for the PSD. The requirement for $S(f)$ to be positive definite implies that:

$$\lambda_{-s} = \lambda_s^*$$

With this property, and defining $z = e^{-\imath 2\pi f \Delta t}$, $S(f)^{-1}$ can be rewritten as a polynomial in z and z^*

$$S(f)^{-1} = \lambda_0 + \sum_{s=i}^{N} \lambda_s z^s + \sum_{s=1}^{N} \lambda_s^* z^{-s}. \tag{4.10}$$

Let z_0 be a root for the polynomial

$$\lambda_0 + \sum_{s=1}^{N} \lambda_s z_0^s + \sum_{s=1}^{N} \lambda_s^* z_0^{-s} = 0,$$

it is easy to show that $(z_0^*)^{-1}$ is also a root for the equation:

$$\lambda_0 + \sum_{s=1}^{N} \lambda s \left(\frac{1}{z_0^s}\right)^* + \sum_{s=1}^{N} \lambda_s^* \left(\frac{1}{z_0^{-s}}\right)^* =$$
$$\lambda_0 + \sum_{s=1}^{N} \lambda_s (z_0^{-s})^* + \sum_{s=1}^{N} (\lambda_s (z_0^s))^* =$$
$$\left(\lambda_0 + \sum_{s=1}^{N} \lambda_s z_0^s + \sum_{s=1}^{N} \lambda_s^* z_0^{-s}\right)^* = 0$$

This means that for every pole z_0 laying outside (inside) the unit circle, there is another pole inside (outside) the unit circle. So, if there are no roots on the unit circle, N roots lay outside and N inside of it.
These properties allow us to rewrite the Fourier expansion (4.10) as [28]

$$S(f)^{-1} = \frac{1}{P_N \Delta t}\left(1 + \sum_{s}^{N} a_s z^s\right)\left(1 + \sum_{s}^{N} a_s^* z^{-s}\right),$$

or, defining $a_0 = 1$,

$$S(f) = \frac{P_N \Delta_t}{\left(\sum_{s=0}^{N} a_s z^z\right)\left(\sum_{s=0}^{N} a_s^* z^{-s}\right)}. \tag{4.11}$$

The MESA expression for the PSD admits a non-linear, all pole representation when evaluated in $z = e^{-\imath 2\pi f \Delta t}$. All the roots of the first polynomial can be chosen to lay outside the unit circle, so that all the roots of the second polynomial will lay inside of it [114]. If one is

able to compute the values of the coefficients a_i (prediction error filter) the PSD is uniquely determined from the previous equation.
The values for the Prediction Error Filter Coefficients and the P_N factor are computed using the Levinson recursion [78] (Reported in Appendix A). It is worth mentioning that, due to Levinson Recursion Structure, the last coefficients a_N are computed by using a very small number of samples from the time-series, and the error in their computation is relevant. In real case scenarios, the sum at the denominator is limited to some order $p < N$, so that the PSD is approximated as

$$S(f) \approx \frac{P_p \Delta t}{(\sum_{s=0}^{p} a_s z^s)\,(\sum_{s=0}^{p} a_s^* z^{-s})}. \tag{4.12}$$

How to determine the best value for p is a non trivial problem and p is the only free parameter of the method. A discussion can be found in [88] An interesting property is that the above expression for the PSD is equivalent to the PSD for an autoregressive process of order p (AR(p))

4.2.1 The Autoregressive process analogy

The application of MESA is not limited to spectral estimates, but it also provides a bridge between spectral analysis and the analysis of autoregressive processes.[114]. An autoregressive process of order P is defined as

$$x_t - b_1 x_{t-1} - b_2 x_{t-2} \ldots b_p x_{t-p} = \nu_t,$$

where b_i are the autoregressive coefficients, and ν_t is a realization at time t of some stochastic process. For our purposes, we assume ν_t to be a white noise process with zero mean and variance σ^2, and we consider the autoregressive coefficients b_i to be real-valued. taking the z[1] transform of the previous:

$$\sum_t x_t z^t - \sum_i b_i z^i \sum_t x_{t-i} z^{t-i} = \sum_t \nu_t z^t = \tilde{\nu}(z) \tag{4.13}$$

and calling $\tilde{x}(z)$ and $\tilde{\nu}(z)$, the transformed quantities, in the z domain, the process takes the form:

$$\tilde{x}(z) = \frac{\tilde{\nu}(z)}{(1 - \sum_{i=1}^{p} b_i z^i)} \tag{4.14}$$

Since we assumed stationarity for the process, $\tilde{x}(z)$ is analytic both on and inside the unit circle. Taking its square value and evaluating it on the unit circle $z = e^{-\imath 2\pi f \Delta t}$, from the definition of spectral density one obtains:

$$S(f) = |\tilde{x}(z)|^2 = \frac{|\tilde{\nu}(f)|^2}{|1 - \sum_{n=1}^{p} b_n e^{\imath 2\pi f n \Delta t}|^2}. \tag{4.15}$$

The numerator is the spectral density of white noise ν_t, i.e. its (constant) variance σ^2.
Comparing (4.15) with (4.12), identifying $b_1 = -a_i$ and $P_N \Delta t = \sigma^2$, the expression for the PSD obtained with MESA is formally equivalent to the PSD of some stationary autoregressive process.

[1] defined as $X[z] = \sum_k x_k z^k$

Having computed the prediction error filter on can easily obtain an estimate of the AR coefficients for the process just changing sign to the prediction error filter a_i. Taken a time series, MESA allows us to compute the PSD and to find an effective description in terms of an autoregressive process. Wold's theorem ensures that this (approximate) description for the phenomenon is legitimate when the process is found to be stationary. So, under the assumption of stationarity, MESA defines a bridge between spectral estimation and the study of autoregressive processes. Another very remarkable result is that the linear predictor obtained solving the error filter equation, is the best linear predictor, i.e. it minimizes error, and is equivalent to a least square fit for the coefficients. [119]

4.3 memspectrum

The MESA method is implemented in a `memspectrum`-python package [89] and fully integrated in pycWB. The package provides a simple interface for the computation of PSDs via MESA, implementing Burg's algorithm for the computation of the prediction error filter.
The package is structured in a modular way, with different files for the implementation of the various parts of the algorithm. The main file is `memspectrum.py`, which contains the main MESA class that implements the MESA algorithm. The class has methods for fitting the model to a time series, computing the PSD, and plotting the results.
The package implements the Levinson Recursion to compute the prediction error filter, but also provides a faster implementation of Burg's algorithm for the computation of the filter coefficients discussed in [119].
The algorithm works with the following steps: firstly, the sample autocorrelation function $\bar{r}$ is computed from the data. The goal of the algorithm is to find the values for the prediction error filter coefficients a_i $(i = 1, \ldots, N)$ and the estimate P_N for the variance of the noise process, with N being the total number of samples in the time-series. The values of a_i and P_N are related to the values of the sample autocorrelation function at various lags (see Appendix A for more details). This solution is recursive—meaning that the coefficient values for the i-th recursion are found from the parameters obtained in the $(i-1)$-th step—and is initialized at step $i = 0$ by setting the initial error power P_0 equal to the signal's variance, represented by $\bar{r}_0$. At the i-th step, only the lags from 0 to $i-1$ of the autocorrelation function are used, i.e., the information brought by larger lags only appears as the recursion progresses. At every step, an estimate for the variance P_i is obtained, together with the values of the a_i parameters and another set of coefficients, b_i, that describe an anti-causal autoregressive process. These are used together with $\bar{r}_{i+1}$ to obtain the estimates for the a_{i+1} and P_{i+1} parameters. In principle, the recursion should be repeated for N steps (the number of data points); however, due to the limited amount of data used in the estimation of the autocorrelation function at the larger lags, the recursion is stopped at an iteration $m < N$. The package implements various features, and is able to fit an autoregressive model of order m to the data, where m_{best} can be found directly by the algorithm implementing a variety of information criteria [20, 70]. Together with the main class, the package fully exploits the autoregressive process analogy, providing methods for forecasting the data, or generate new synthetic time series based on a given PSD.
The package is documented with docstrings and examples, and is designed to be user-friendly and easy to use. The code is presented in [88] together with some examples of application in CBC parameter estimation, especially in the context of GW150914 analysis.

Chapter 5

Whitening in cWB: MESA Algorithm and wavelet tuning

5.1 Introduction

From this chapter onward, the thesis shifts from a theoretical perspective to a technical one, displaying the results of the work carried out during the PhD. We open the chapter with a short discussion on the wavelet transform that is at the core of the procedure. The goal of this short section is to show that the standard tuning of the wavelet transform in cWB and pycWB has been prioritizing spectral performances versus timing resolution, and that by tuning the parameters of the transform it achieves a relevant improvement in terms of compactness in time-frequency representation, approaching the Heisenberg-Gabor limit. Despite being simple, this new tuning has an overall positive impact on the quality of the reconstruction of GW transients.
The chapter then focuses on our investigations on data whitening procedures. Data whitening is a standard procedure in time series data analysis and is ubiquitously implemented in GW analysis pipelines. Nevertheless, its impact on the data has some subtleties that are not always fully appreciated and are not usually mentioned in the standard literature.
Section 5.3.1 provides a theoretical description of the procedure in terms of linear filter theory and discusses how the whitening procedure has been implemented in pycWB. In particular, we focus on two types of whitening filters that can be obtained using MESA: the "autoregressive" and the "PSD" whitening. Finally, we compare the results obtained with the MESA whitening procedure with the standard pycWB procedure, and discuss the performances both on pure noise and on signals. Special attention is devoted to time-leakage effects, i.e. the contamination from high SNR signal parts (e.g. from the merger in CBC transients) into neighboring times (e.g. ring-down of the remnant). In general, this is relevant for time-resolved spectroscopic studies and impacts also searches for post-merger echoes from black hole mimickers [91, 44].

5.2 Approaching the Gabor-Uncertainty Limit

This section discusses the tuning of the WDM transform [92, 47], introduced in 3.1. The WDM transform is a wavelet transform that provides a complete basis to represent time series in terms of time-frequency pixels. It is built in such a way that the basis function

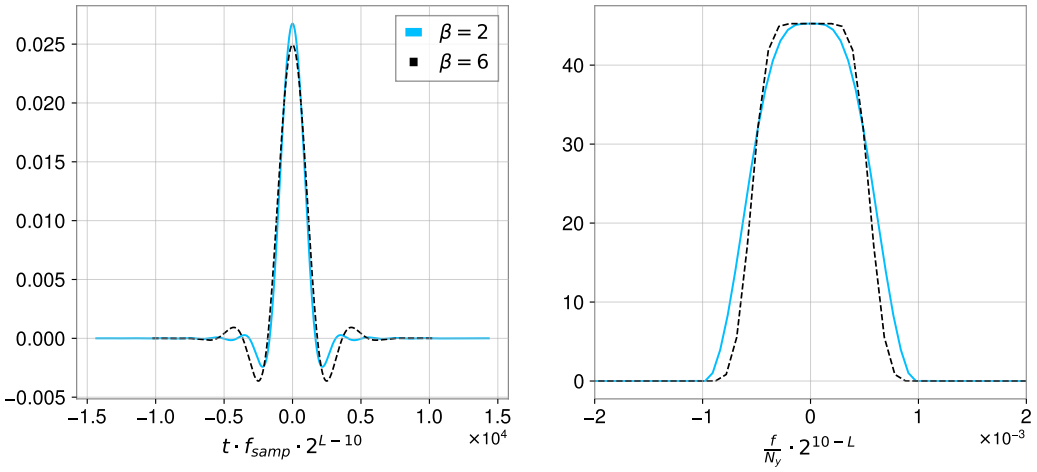


Figure 5.1: The WDM envelopes in time (left) and frequency (right) as a function of the β order. In the plot, f_{samp} is the sampling frequency, N_y is the Nyquist frequency and L is the resolution level for the analysis.

is compact in the frequency domain, while exhibiting unbounded tails in the time domain, that are then truncated in the algorithm. Nevertheless, the shape of the basis function is not completely fixed, but can be tuned by changing the values of some parameters. The two most important parameters are the resolution level, which acts as a reciprocal scale factor in the two domains, and the β order, which controls the shape of the wavelets' envelope. The cWB and pycWB pipelines use different resolution levels simultaneously, to capture different time-frequency scales, and their impact is be discussed in the next chapters. The effect of the β order on the wavelet basis is shown in Figure 5.1. It controls the "trade-off" between the precision of the transform in the frequency domain and the time domain. Larger values for β are associated with a better frequency resolution and vice-versa. So far, cWB and pycWB analyses have been using a default value of $\beta = 6$, and in this work we investigated alternative options.

From a pure whitening perspective, the default value of $\beta = 6$ should be preferred since it provides the best precision in frequency domain, but at the cost of a degraded resolution in time. Finding the best compromise between the accuracy of the representation in the two domains is doable, and it can be expressed in terms of the distance of the transform from the "Gabor Uncertainty Limit" [59]. This corresponds the Heisenberg Uncertainty principle applied in the context of signal processing, and it states that there is a lower bound for the product of the uncertainty over time and the uncertainty over frequency of a signal, which is given by:

$$\sigma_t \sigma_f \geq \frac{1}{4\pi}, \tag{5.1}$$

with the uncertainty over time and frequency defined in equation 3.13. The tuning of the WDM which minimizes the product $\sigma_f \sigma_t$ provides the best trade-off between the resolution in time and frequency domains. Since the WDM wavelets are hard to treat analytically, we compute the uncertainty numerically, by computing the time and frequency envelopes of the wavelet basis and then computing the corresponding uncertainty as defined above. The results are shown in Figure 5.2 as a function of the β order and resolution level. The left panel shows the Gabor Uncertainty limit, while the center and right panels show the time

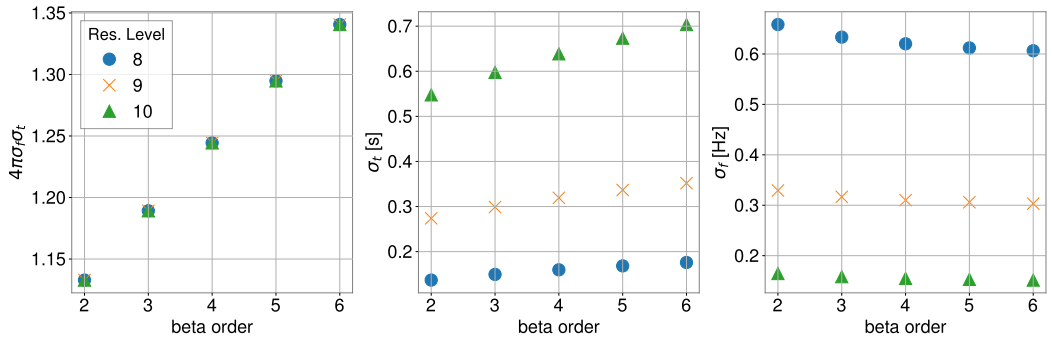


Figure 5.2: The Gabor Uncertainty Limits for the WDM transform (on the left), the time uncertainty (center) and the frequency uncertainty (right) as a function of the β order and resolution level, compute for a sampling rate of 2048Hz

and frequency uncertainty, respectively.
On the left panel, we can see that the Gabor Uncertainty limit is always above the lower bound given by the Gabor Uncertainty, but its distance to the limit decreases monotonically with the β order, with $\beta = 2$ providing an improvement of 20% when compared to the standard value $\beta = 6$, meaning than we obtain an optimization of the time-frequency representation by moving from $\beta = 6$ to $\beta = 2$. For the value $\beta = 1$, the time domain envelope is $\propto t^{-2}$, hence its variance diverges and the Gabor Uncertainty limit is not defined. In the following analysis, we assess the impact of the wavelet tuning on both the data whitening and the waveform reconstruction of the pycwb pipeline.

5.3 Whitening via MESA

This section focuses on the construction of an algorithm that uses MESA as the core estimator to design a new whitening procedure, different from the built-in version that pycWB inherited from cWB. We first characterize the whitening procedure in terms of linear filter theory, showing that MESA can be used to build two different types of whitening filters: one based on the autoregressive analogy and one based on the PSD.
We then develop a whitening algorithm fully compatible with pycWB, with a strong focus on making the estimate robust against glitches or signals in the data. The whitening transformation is a linear filter that takes as an input a time series and output a new time series with a flat PSD, i.e. white noise. Our goal is to characterize the whitening procedure in terms of linear filter theory and determine the transfer function of the whitening filters built using MESA.
Let $x(t)$, $\omega(t)$ be two one dimensional time series, the function $T : \mathbb{R} \to \mathbb{R}$ that maps the input time-series $x(t)$ to the output whitened time series $\omega(t)$ can be described as a linear filter. For such a filter, the input and output timeseries are related via the **transfer function T(t)**. The output time-series is is obtained as the convolution between the transfer function and the input time series:

$$\omega(t) = T(t) * x(t) = \int_{-\infty}^{\infty} T(t-\tau)x(\tau)d\tau = \int_{-\infty}^{\infty} T(\tau)x(t-\tau)d\tau \tag{5.2}$$

or, for discrete signals:

$$\omega_t = T_t * x_t := \sum_{\tau=-\infty}^{\infty} T_{t-\tau} x_\tau = \sum_{\tau=-\infty}^{\infty} T_t x_{t-\tau}. \tag{5.3}$$

In case of linear stationary filters, the functional form of T uniquely determines the filter's behavior. In Fourier domain, the convolution becomes a simple product of the transformed function, i.e.:

$$\tilde{\omega}_f = \tilde{T}_f \tilde{x}_f \tag{5.4}$$

and $\tilde{T}$ describes the frequency response of the digital filter. In general $\tilde{T}$ is a complex function and is characterized by its norm and phase:

$$\tilde{T}_f = |\tilde{T}_f| e^{i\phi_f}. \tag{5.5}$$

In the next section we interpret whitening transform as linear filter, retrieving the functional form of the transfer function.

5.3.1 Autoregressive VS PSD whitening

Whitening via an Autoregressive model

As we have seen in Section 4.1, MESA provides an estimate of the PSD in terms of an AR(p) process, determining its coefficients a_i and the variance of the noise σ_ν^2. In this representation, the time series at time t is represented as a linear combination of the previous p values of the time series plus a noise term. In this way, the time series at times t can be written as:

$$x_t = \sum_{i=1}^{p} a_i x_{t-i} + \sigma_\nu \omega_t^{ar} \tag{5.6}$$

with ω_t^{ar} white noise drawn from a normal distribution with zero mean and unit variance. Defining $a_0 = b_0 = 1$ and $b_i = -a_i$ for $i = 1, \ldots, p$, one finds

$$a_0 x_t + \sum_{i=1}^{p} (-a_i) x_{t-i} = \sum_{i=0}^{p} b_i x_{t-i} = \sigma_\nu \omega_t^{ar}, \tag{5.7}$$

So that one can construct a white-noise time series using the autoregressive coefficients a_t:

$$\omega_t^{ar} = \sum_{i=0}^{p} \frac{b_i}{\sigma_\nu} x_{t-i} \tag{5.8}$$

In this case the transfer function can be easily written in time domain:

$$T_t^{ar} = \begin{cases} b_t/\sigma_\nu & \text{if } t \in [0, p] \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

So, at every moment, the output time series is built as a linear combination of the previous p-elements. Hence the filter $T(t)$ is **non symmetric** and preserves causality.
The functional form in frequency domainis case-dependent and is related to the prediction error filter coefficients b_t. In general, since T^{ar} is non-symmetric and real, we can conclude that $\tilde{T}_f$ is hermitian

$$\tilde{T}_f^{*ar} = \tilde{T}_{-f}^{ar} \tag{5.10}$$

and, in general:

$$\tilde{T}_f^{ar} = \frac{1}{\sqrt{N}} \sum_{m=-\infty}^{\infty} T_m^{ar} e^{-2\pi i f m \Delta t} = \frac{1}{\sqrt{N}} \sum_{m=0}^{p} \frac{b_m}{\sigma_\nu} e^{-2\pi i f m \Delta t} = \mathscr{F}\left[\frac{\mathbf{b}}{\sigma_\nu}\right] \tag{5.11}$$

with $\mathbf{b} = \{b_0, b_1, \ldots, b_p\}$ the vector of the prediction error filter coefficients.

Whitening via Power Spectral Density

In general, when dealing with a wide-sense stationary stochastic process of infinite duration, the noise can be assumed to follow a Normal distribution with a covariance matrix that is diagonal in the Fourier domain. The diagonal elements are given by the PSD of the process. This is a standard assumption in GW analysis, where detector noise is typically modeled as wide-sense stationary and Gaussian, requiring appropriate windowing of the data to take into account finite-duration effects. Once the PSD is estimated using appropriate methods, the noise time series in the Fourier domain is described by the **Whittle Likelihood** [103].

$$p(\tilde{x}_f) \propto \exp\left\{\sum_f \frac{\tilde{x}_f^2}{S_f}\right\} \tag{5.12}$$

which is a normal distribution with zero mean and diagonal covariance matrix. By simply defining

$$\tilde{\omega}_f^{psd} = \frac{\tilde{x}_f}{\sqrt{S_f}}, \tag{5.13}$$

it's easy to show that $\omega_t \sim \mathcal{N}(0,1)$, i.e. white noise. By comparing the above equation with 5.4, the transfer function in the Fourier domain takes the form:

$$\tilde{T}_f = \frac{1}{\sqrt{S_f}}. \tag{5.14}$$

Since $S(f)$ is real valued, $\tilde{T}_f$ is real, meaning that the phase response of the filter is 0 whatever frequency:

$$\phi_f = 0, \tag{5.15}$$

while the Gain is just one over the square-root of the PSD, making the PSD of the output process flat.
In general, in time domain we can then write:

$$\omega_t^{psd} = T_t^{psd} * x_t = \mathscr{F}^{-1}\left[\frac{1}{\sqrt{S_f}}\right] * x_t \longrightarrow T_t^{psd} = \mathscr{F}^{-1}\left[\frac{1}{\sqrt{S_f}}\right] \tag{5.16}$$

The above is the most general form for the transfer function $T^{psd}(t)$ used when the time series is whitened via amplitude PSD division. Since the PSD is real valued, positive-definite and symmetric $T^{psd}(t)$ is itself real-valued and symmetric. Time symmetry, together with the definition in 5.3, implies that, for every t, ω_t is built mixing every past and future data, since T_t is non-zero for $t < 0$. Therefore such a filter does not preserve causality.
Taking advantage of the mesa-autoregressive analogy 4.2.1, we can write an explicit form of T_t^{psd} in terms of the autoregressive coefficients. Remembering that:

$$S_f = \frac{\sigma_\nu^2}{|\mathscr{F}[\mathbf{b}]|^2} \tag{5.17}$$

we obtain:

$$T_t^{psd} = \mathscr{F}^{-1}\left[\left|\mathscr{F}\left[\frac{\mathbf{b}}{\sigma_\nu}\right]\right|\right]. \tag{5.18}$$

Comparing Equations (5.9) and (5.18) one finds that, in the Fourier Domain, the autoregressive and the PSD filter are related by a very simple relation

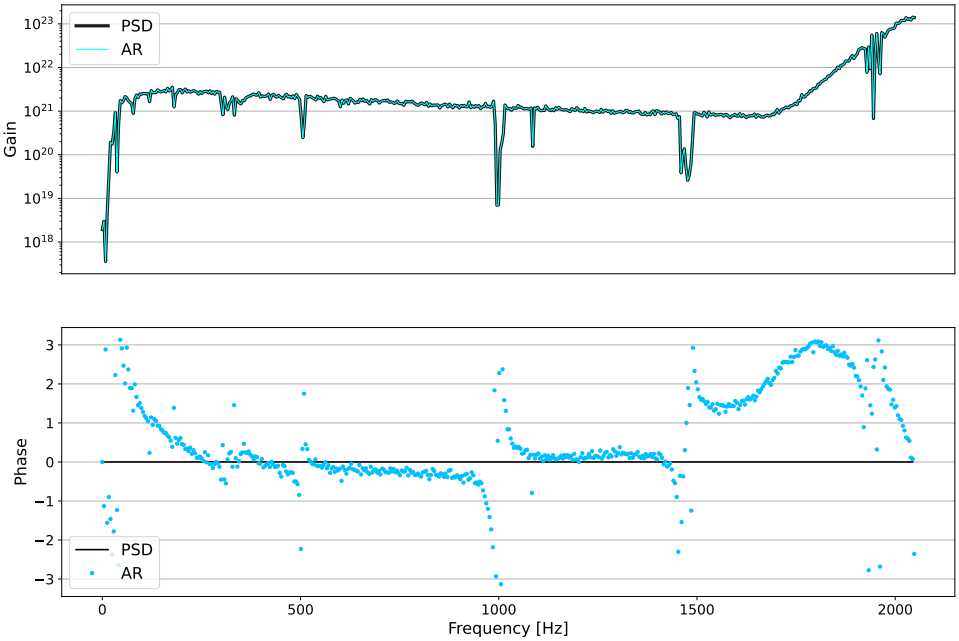


Figure 5.3: Bode Plot for the two filters built using the PSD estimate and the autoregressive process analogy. The gain and the phase response of the filters are built by applying the whitening filters on sinusoids of different frequencies and comparing the amplitude and the phase of the input and the output. The Gain of the filters is almost identical, while the phase response are highly different. The PSD method has 0 phase response for every frequency, while AR analogy has a complex pattern

$$\tilde{T}_f^{psd} = |\tilde{T}_f^{ar}|.$$

This means that the two filters are very similar, They have, by definition, the same Gain in frequency domain, but every information about the phase is completely lost in T^{psd}.
Also, notice that, having the same Gain, due to Wiener-Khinchin theorem, the two filters have the same energy:

$$E^{ar} = \sum_f |\tilde{T}_f^{ar}|^2 = \sum_f |\tilde{T}_f^{psd}|^2 = E^{psd} \tag{5.19}$$

In Figure 5.3 we show the Bode plot built for the two response functions T^{ar} and T^{psd}. The PSD and AR coefficients are retrieved by computing MESA PSD estimates on ~ 11 seconds of data at the beginning of the O1 public data [33][8] containing GW150914 [7] at a sample rate of $4096Hz$. The input signals are 500 sinusoids each with a different frequency linearly spaced within the range $[0, 2048]Hz$. These signals are then whitened and T is computed

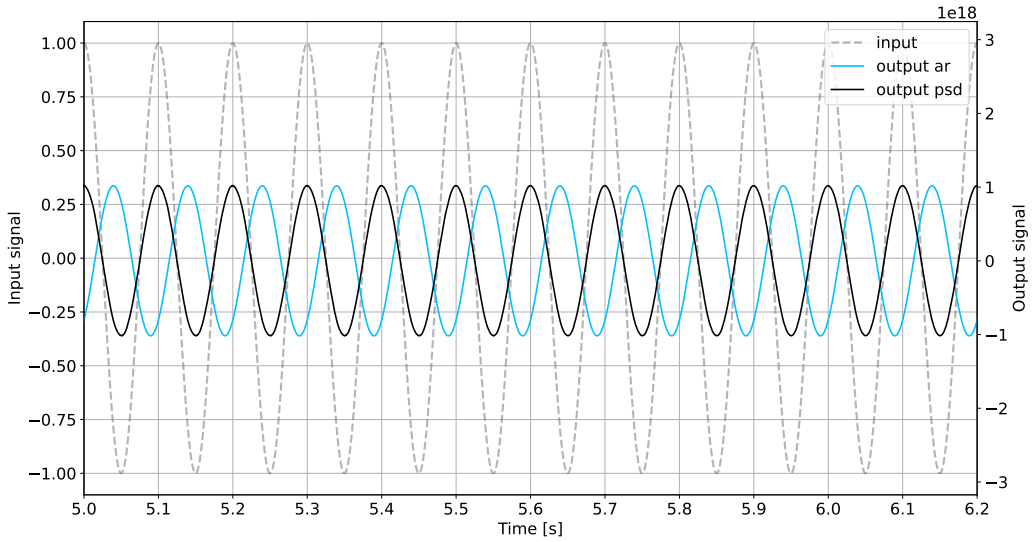


Figure 5.4: Effect of the different whitening filters on one sinusoid with 10 Hz Frequency (gray dashed line). The PSD-filtered output (black, solid line) displays a change in absolute magnitude but no phase shift in the data. The autoregressive filter (blue, solid line) shows the same gain of the first filter together with a phase shift of the signal

inverting Equation 5.4.

The Frequency gain of the filter is identical in the two cases, and the two lines are superimposed. The phase response of the filters is very different in the two cases. For PSD, as we demonstrated before, it has 0 response across the whole frequency range.
The phase response for the AR filter is instead non-linear. Nevertheless, by construction the MESA filter is a minimum-phase, finite Impulse Response (FIR) filter [31]. Under these conditions, Bode showed that the phase and amplitude response of the filter are related by the Bode Gain-Phase relation [30]:

$$\phi(f) \simeq \frac{\pi}{2} \frac{d \log |\tilde{T}(\imath f)|}{d \log(f)}, \tag{5.20}$$

so that the phase is larger where the gain is steeper, and it is flat where the gain is flat. An example of the actions of the filter on one sinusoidal signal with $10Hz$ frequency is reported in Figure 5.4, where both the gain and frequency response of the two filters discussed are clearly visible.
When analizing the data, since beside the detection we also care about the reconstruction of the signal, the phase response of the filter is an important aspect to take into account. A non zero-phase response can cause a distortion of the signal morphology, which can be problematic for an accurate reconstruction of the signal. We now want to investigate the impact of both the whitening filters on the morphology of signal that may be present in the data.

AR and PSD whitening on signals

In the examples so far, we have limited our discussion to pure noise or signals that represented by perfect sinusoids with no noise and ”infinite” duration. Here we treat the case of a linear

superposition of noise and signal

$$d(t) = h(t) + n(t) \tag{5.21}$$

with d the data, n the noise realization and h some signal. Thanks to the linearity of the whitening filter, in absence of glitches, the above equation in the whitened domain is simply:

$$d_w(t) = h_w(t) + \omega(t). \tag{5.22}$$

In this section we want to show what's the effect of the whitening filter on the signal, i.e. how h_ω and h are related. For this reason, we work with "zero noise" signals. With zero-noise signal we mean that we treat separately the noise and the signal. The former is used to build the whitening filter, while the latter is used to test the effect of the whitening procedure on signals without noise. Under these assumptions, the "zero noise" signals can be interpreted as

$$h_\omega(t) = d_\omega(t) - \omega(t).$$

Clearly, such a separation can only be performed in a simulation environment, in which the signal and the noise are known separately, but not in a real analysis, where the signal is unknown and the noise is only known through the data. However, these assumptions are not only a good exercise to better understand the effect of whitening procedure, but, due to the relative low SNR of the signals and the mitigation of signal's presence in the whitening procedure (we discuss it later on in this chapter), it represents a realistic scenario for the kind of signals detected so far [1]. In real analyses, is impossible to carry out the "noise subtraction" from the data in this simple way.

[1] This is probably not the case for next generation detectors such as Einstein Telescope, which will be signal-dominated detectors rather than noise-dominated.

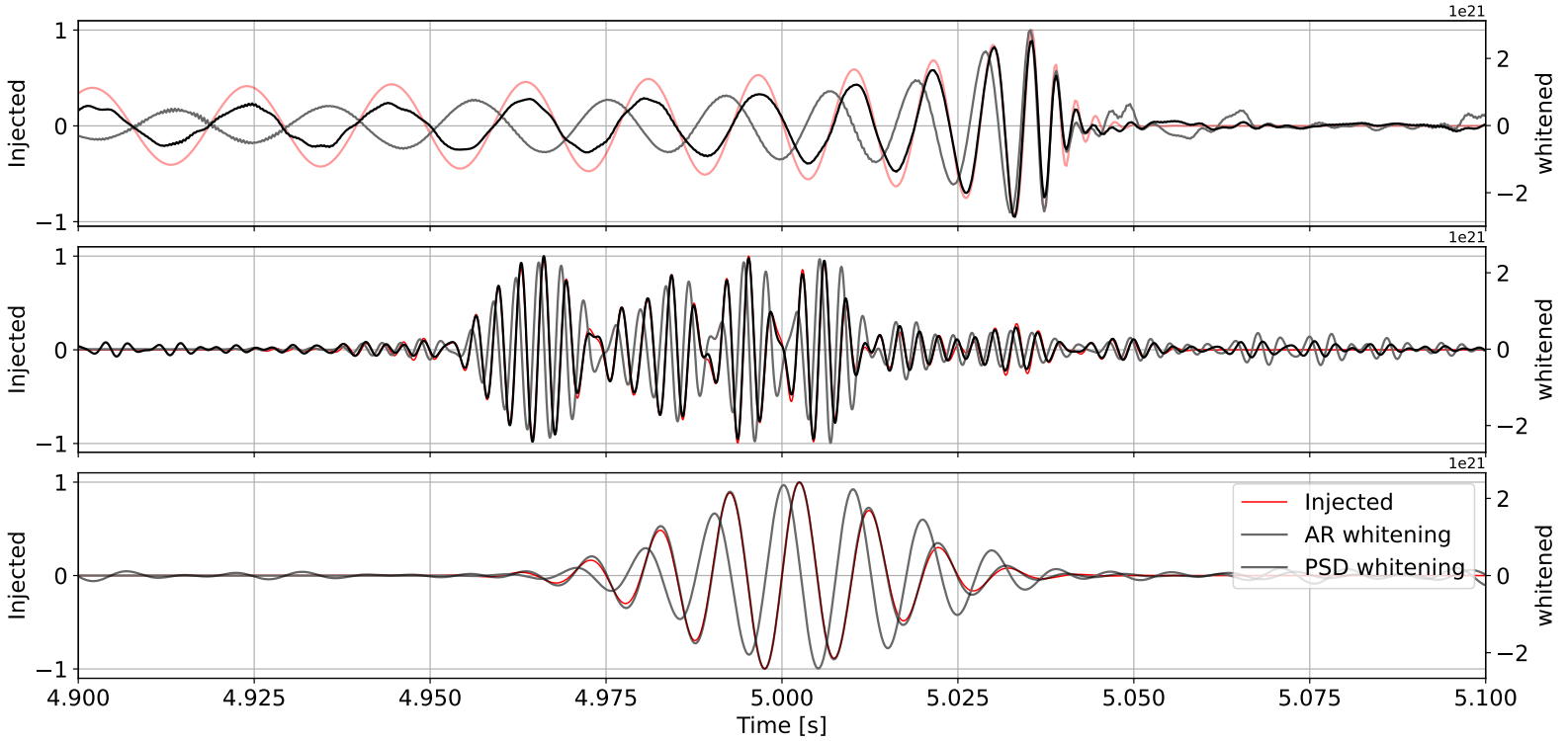


Figure 5.5: Effect of whitening filters on three different waveforms. From top to bottom to right: a CBC-like signal, a White noise Burst with central frequency $f_0 = 100Hz$ and bandwidth $\Delta f = 50Hz$, a sine-Gaussian with central frequency of $100Hz$ and quality factor $Q = 9$. The red line represents the simulated signal, the black line represents the whitened signal with the autoregressive filter and the blue line represents the whitened signal with the PSD filter. Both filters show the same gain, with PSD filter only preserving the phase. In the CBC plot different frequency content display different gain. In both cases a temporal leakage of the signal is present, one-sided (causal) for the AR filters and symmetric (non causal) for PSD filter

In Figure 5.5 we can see that both the effects on Gain and Phase of the filter are reproduced. The two filters display the same frequency-gain, as it is particularly evident in the CBC-like signal case.The phase shift is only present for AR whitening, while PSD whitening preserves the phase content.
Moreover, another phenomenon emerges: whitening the data causes a leakage of the whitened signals in regions adjacent to the signal where no signal amplitude was originally present.

Time Leakage

The time leakage shown in figure 5.5 is a direct consequence of the convolution between the input signal and the transfer function of the filter, and is an intrinsic phenomenon of the whitening procedure. In fact, the time domain representation of the linear filter as a matrix T_{ij} is in general non diagonal. T_{ij} would be diagonal if the noise process was already decorrelated, so that there would be no need to whiten the data.
This means that, in general, the output signal at time t is a mix of the "neighboring" time components of the input signal properly modified by the transfer function. The number of time components that are mixed together depends on the time-length of the filters, i.e. how many non diagonal components of the filter are different from zero. This means that, whatever whitening filter is used, the leakage over time is an intrinsic feature of the whitening procedure, and it is not possible to avoid it.
Looking back at Figure (5.5), we can notice that for AR whitening the Leakage is only found **after** the end of the signal. This is due to the causal behavior of the filter. In fact, every white sample is built as a linear combination of the previous p samples. Hence, the Leakage can only happen after the signal, not before.
On the contrary, time leakage is found both before and after the signal when the data are whitened using the "PSD" filter. Again, this is due to the fact that this filter does not preserve causality, hence at every time t the whitened array is built as a combination of the time series at every time, allowing the signal to also contaminate the output time-series backward in time.

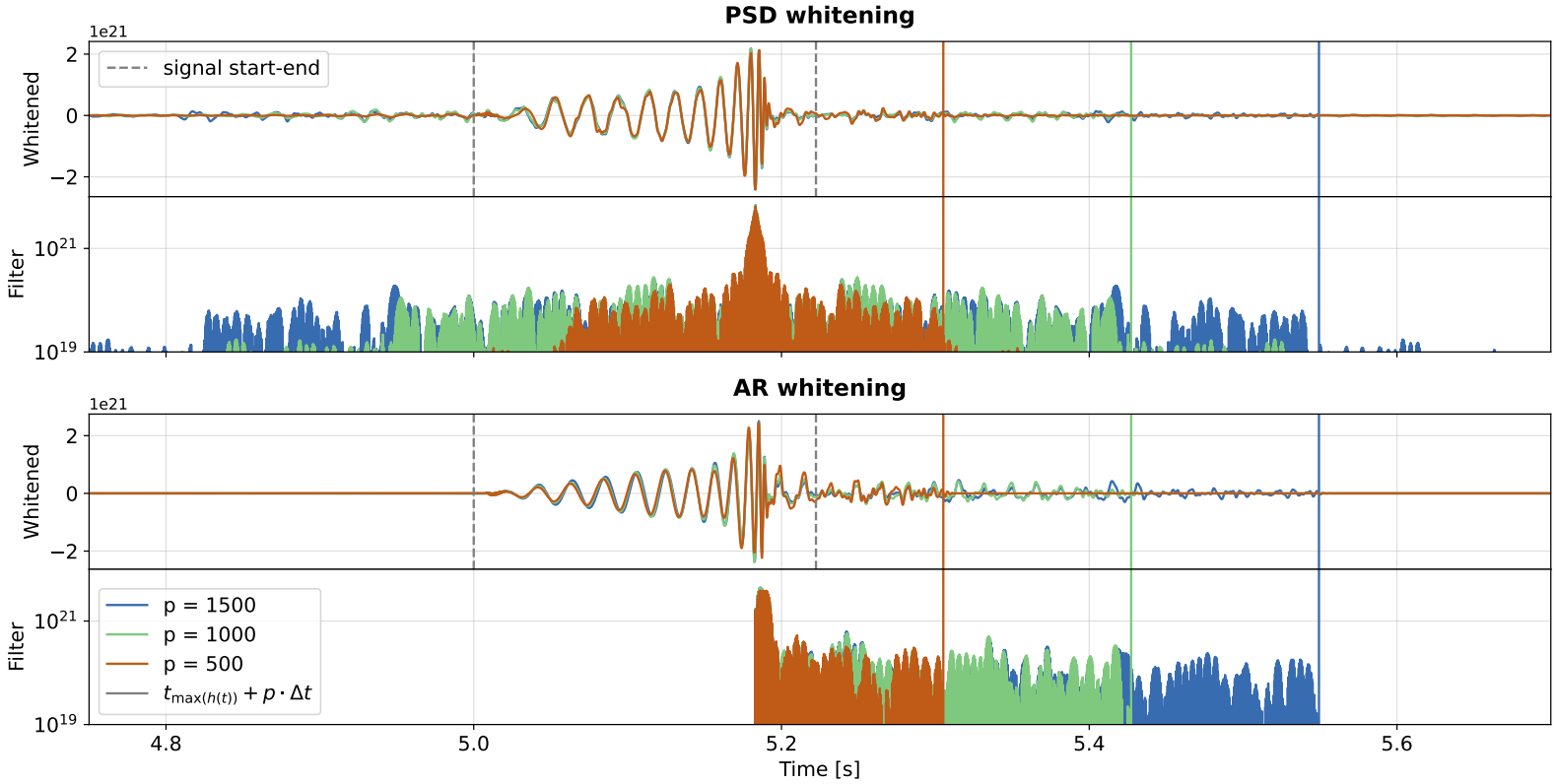


Figure 5.6: Time leakage of the signal after whitening procedure, for both autoregressive and PSD filters. The top panel shows the whitened signal using the PSD whitening for the three autoregressive orders of $m = 500$ (red), 1000 (green), 1500 (blue). The gray, dashed lines represent the start time and the end time of the generated strain signal.
The central-top panel shows the whitening filter in time domain, centered on at the maximum amplitude of the signal. The three vertical lines represent the distance in time from the maximum of the signal to the time associated to the autoregressive order m of the filter, meaning $\tau_m = m\Delta t$. The bottom panel shows the same plots for the autoregressive whitening filter. The time leakage is present in both cases, but it is one-sided for the AR filter and two-sided for the PSD filter. The duration of the leakage increases with the autoregressive order, and it is related to the time-length of the filter.

The role of the autoregressive order

The different ways in which the two filters cause time leakage is related to their causal and non-causal nature. We said that the time leakage is an intrinsic feature of the whitening procedure, but with MESA one can tune the duration of the time leakage by changing the value of the autoregressive order p of the associated AR process. In fact, as we said, the leakage is related to the time-length of the filter, see eq. 5.9. For the PSD filter drawing such a conclusion is not as straightforward.
To show how the length of the leakage is related to the autoregressive order, we whiten the previous CBC-like signal with both AR and PSD filters built with 3, different fixed auto regressive orders of 500,1000 and 1500 respectively. The results are plotted in Figure 5.6: top plots show the whitened signals using both filters, bottom plots show the corresponding whitening filters in the time domain. The autoregressive order increases from left to right. As the autoregressive order increases, the duration of time leakage increases accordingly. While for the AR filter this is easily explained by its construction, also the PSD filter drops to negligible values outside a larger time interval, whose right end coincides with the same duration of the auto-regressive process upon which the PSD estimate is built.

A second effect, unrelated to the autoregressive order, is the different distribution of the energy of the leaked signal between the two filters. In fact, on the plot it is easily seen that for $\tau > 0$, the PSD filter takes smaller values, resulting in a lower amplitude leakage after the signal. However, the total energy of the two filters is the same by construction (see Equation [5.19]), i.e. the PSD filter is just diluting the overall energy leakage in a wider time interval that extends also in $\tau < 0$.

The above discussion shows that the two filters display some similarities but act on the signal in different ways.
The choice between the two filters strongly depends on the specific application. If we only wanted to work on detection problems, both filters could be used. But, since burst analysis is not only focused on detection but also on the reconstruction of the signal (and this thesis is devoted to this), the phase response of the filter is an important aspect to take into account. For this reason, the PSD whitening is the best choice for our purposes, since it preserves the overall frequency and phase content of the signal[2].

5.4 Building a robust whitening procedure with MESA

Having discussed the different types of whitening filters that can be built using MESA, we now want to build a whitening procedure that can be implemented in pycWB. This procedure based on the PSD whitening, needs several precautions to take into account the general properties of noise. pycWB works on data segments of duration that can go up to 1200 seconds. For such durations, the noise process cannot be considered as stationary, so that estimating the PSD at once on the whole segment, and use it to whiten all the data in the segment, is not a good choice. For this reason, the whitening procedure is implemented on shorter sub-segments of data, so that the PSD estimate is more representative of the noise process.
The whitening procedure is then implemented as follows: The data are divided in subsequent,

[2]For other applications, one could also consider the use of the AR filter, built as causal or anti-causal filters. This could be useful in the pure detection of precursor or post-merger signals, in which the possibility to have a time leakage only in one direction can avoids dangerous SNR leakage in the relevant time window.

overlapping windows of 15 seconds, separated by 5 seconds[3], even if different values for the window length and separation can be chosen by the user. For each 15 seconds window a PSD estimate is computed, so that a different PSD estimate is obtained every 5 seconds. To increase the robustness of the estimate, the user can decide to take the median of $2n+1$ PSDs estimate centered around the current window, with n a parameter that can by chosen in the configuration file. The median reduces the statistical fluctuations and mitigate the effect of the outliers on the estimate. Notice that, for standard periodograms, the median is not an unbiased estimator of the mean, so that a correction factor is needed to get an unbiased estimate of the PSD [21]. For MESA instead, the realizations are symmetrically distributed around the mean [88], so that the median can be used as a robust estimate of the mean without needing any correction factor.
Once that a sequence of PSD estimates is obtained, an isolation forest algorithm is applied to assess and mitigate the presence of outliers. This procedure is discussed into more details in section 5.4.1. In case a glitch affects a PSD estimate, a neighbor estimate is chosen in its place.
We now end up with several PSD estimates, each associated with a specific window of the data. Each window is then whitened using the associated PSD estimate.
Since the whitening procedure is implemented in overlapping windows, only the central 5 seconds of the data are kept, and the rest are discarded. This procedure of keeping the central seconds only is necessary because whitening procedures are affected by boundary effects at the edges of the window, so that data close to the edges must be discarded. With this procedure, the boundary effects are only present at the very beginning and at the very end of the job segment, while for the rest of the data, the whitening procedure is not affected by boundary effects. It is important that the stride between two consequent windows is smaller or equal to one third of the window length, so that there are no discontinuities in the whitened data. Smaller values are not necessary because they would require more steps. This procedure should be able to whiten the data in a robust way, taking into account the non-stationarity of the noise process and mitigate the effect of glitches on the PSD estimate. A scheme of the procedure is synthesized in Appendix B.

5.4.1 Sensitivity to Glitches

We now discuss in more detail the effect of glitches on the whitening procedure and how to mitigate their impact on the PSD estimate. PSD estimates are sensitive to data transients, including both instrumental glitches and genuine signals. Even modest gravitational-wave (GW) signals can contaminate the PSD estimation and suppress the recovered signal amplitude. This effect scales with signal strength and is particularly pronounced for loud events like GW250114 [3] (Network SNR ~ 70). Consequently, employing a rolling median of the PSD estimates is a standard way to mitigate this effect. While taking the median is usually sufficient, there are some corner cases for which it is not enough to have a robust PSD estimate. Such cases include situations like the presence of more than one glitch separated by a relatively short time interval, or multiple signals close in time. In these situations, several consecutive PSD estimates can be affected by the transients, meaning the median would have to be taken over a very large number of PSD estimates. While this is doable in principle (and in our algorithm, the number of PSD estimates used for the median is a parameter that can be set by the user), it's not easy to find a general tuning that can deal with all possible scenarios. For this reason, we want to propose another approach to

[3]The standard choice of cWB are 60 and 20 seconds respectively for the window and the stride 3.2.1. The values can be configured by the user.

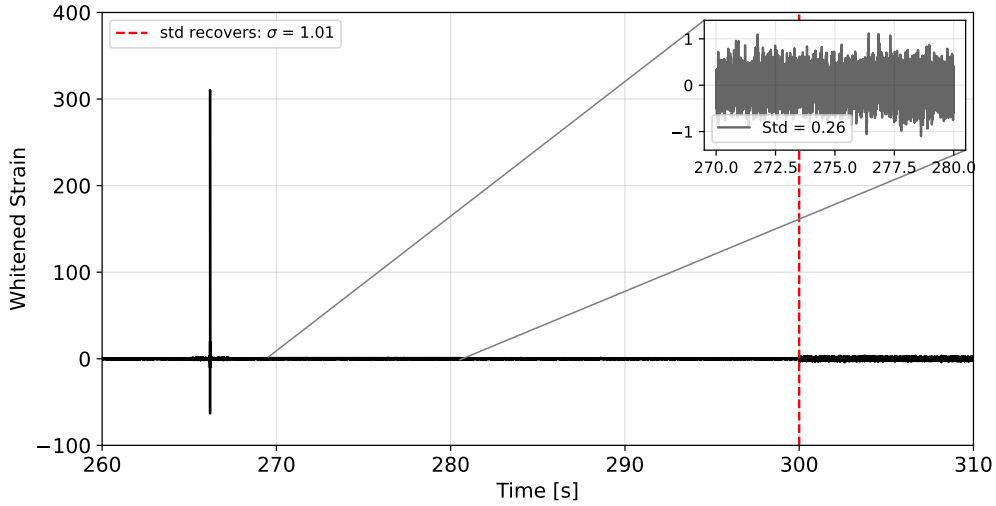


Figure 5.7: Example of the impact of a glitch on whitened data. The PSD estimate is biased by the glitch's presence for $\sim$ 40 seconds and only recovers when a new PSD is computed. The bias causes a significant reduction of the variance of whitened data.

mitigate the impact of transients on the estimate of the whitening filter. An example of the impact of a glitch on the whitening procedure is shown in Figure 5.7. The displayed glitch is found after 266.2 seconds from the start of the job segment, and its presence is found to completely alter the estimate for the PSD, hence the functional form of the whitening filter. As we can notice, for about 40 seconds of data following the glitch, the noise is highly affected and its standard deviation is reduced to $\sigma = 0.26$. Starting from 300 seconds, we are far enough from the glitch and noise recovers the expected behavior.
This is done to the an overestimation of the PSD due to the presence of the glitch (Figure 5.8), which causes a reduction of the gain of the filter, hence reducing the overall power of the output white noise. In Figure 5.8 we show the impact of the glitch on the PSDs. The red lines show the estimate of the PSDs on the data segment when no glitch rejection is implemented, so that the PSD estimate is contaminated by the glitch. The black lines show the variability of the PSDs over the same job segment when the possible presence of glitches is addressed. We can notice that, for loud glitches, they may affect the estimate of the PSD in the whole frequency range, increasing the estimated power of the noise energy with respect to the standard variability. Some other weaker glitches may only affect a narrower frequency range, but still with a non-negligible impact on the estimate.
Since whitening is a procedure that we want to implement on the whole dataset, and process several chunks of data for this purpose, it is is important to find a fast and reliable way to mitigate the impact of loud glitches on the PSD.

5.4.2 Glitch Rejection as an outlier Detection Problem

The problem of the glitch rejection can be interpreted as an outlier detection problem. This means that we expect glitch to be rare events in the data, compared to the total duration of the data, so that we can detect them as outlier in the distribution of the noise energy.
The idea is the following: we build a Glitch-Classifier G such that it values **True** if the

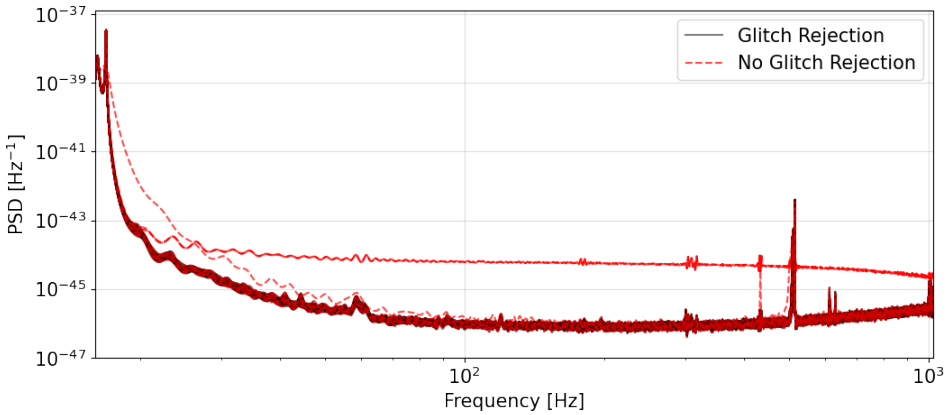


Figure 5.8: Impact of a glitch on the PSD estimate. The red line shows the estimates of the PSDs without the glitch rejection while the black lines show the PSD estimate with the rejection of glitch contaminated PSDs.

estimate is supposedly affected by a glitch, and **False** otherwise. Whenever G is True, the corresponding estimate of the PSD is then substituted with the **closest in time** estimate for which G is False.
In this way we aim to an estimate which is as representative as possible of the noise process and, being the closest in time, still able to correctly represent the local properties of the process despite nonstationarities.

The rejection is implemented using an unsupervised learning method for outliers detection named **isolation forest**[80], which is presented in Appendix C.1.

Discarding the outliers, building the features

The Isolation Forest algorithm does not work with time series, but requires a set of features to make inference on. To build the features we compute a set of statistics that capture the deviation of each PSD estimate from a reference value, i.e we try to quantify the excess of energy in the noise and when it can be considered as an outlier. The excess has to be referred with respect to some other quantity, which is suppose to be the "real" PSD for un-glitched, pure noise process. Since such a quantity is not available and since we don't know a priori where a glitch is found, we have to build the reference PSD from the estimates obtained on the same job segments. Due to its robustness, we use as a reference the median $\bar{S}_f$ of all the PSD estimates obtained on the job segment. For every PSD estimate, the deviation r of the reconstruction from the reference is computed as:

$$r = \sqrt{\frac{1}{N_f} \sum_f \left[\log\left(S_f\right) - \log\left(\bar{S}_f\right)\right]^2}. \tag{5.23}$$

Among other various possible statistics this has been chosen because it has shown to maximize the separation of the glitches with respect to the central-cluster of healthy estimates. The logarithm of the PSD has been chosen because we have two varying scales: the one of the PSD, which span several different order of magnitudes, and the noise fluctuations of the PSD estimate. To suppress the latter and make the former dominant, the logarithmic metric is a good choice. To increase the sensitivity across different frequency ranges, the

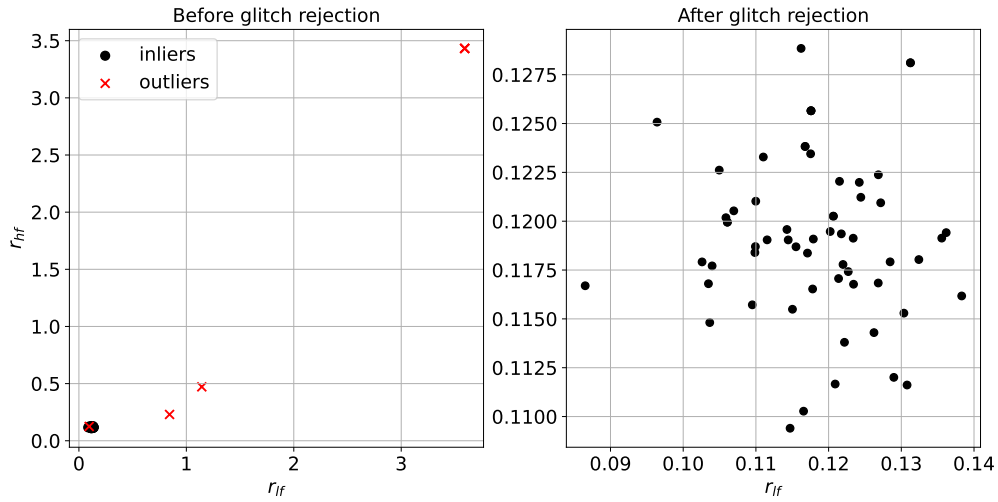


Figure 5.9: Scatter Plot of the features r_{lf} and r_{hf} (Equation (5.24) used to classify the presence of glitches in the data. On the left side, the distribution before glitch rejection is implemented, on the right side, the distribution after glitch rejection is implemented

summation is split in a "low frequency" and a "high frequency" range, which are defined as:

$$r^i_{lf} = \sqrt{\frac{1}{N_f}\sum_{f=16}^{f_{cut}}\left[\log\left(S^i_f\right) - \log\left(\bar{S}_f\right)\right]^2}; \qquad r^i_{hf} = \sqrt{\frac{1}{N_f}\sum_{f=fcut}^{f_{Ny}}\left[\log\left(S^i_f\right) - \log\left(\bar{S}_f\right)\right]^2} \tag{5.24}$$

with the lower threshold of 16Hz the standard lower frequency cutoff for pycWB analysis and with f_{Ny} the Nyquist frequency. The threshold f_{cut} has been chosen as the value that maximizes the separation between outliers and "inliers" after testing several different values on a large number of data segments, and has been set to $f_{cut} = 128$Hz.
The outliers are then classified by the application of an Isolation Forest algorithm in which each data is represented by the pair of features (r_{lf}, r_{hf}). The output of the Isolation Forest is:

$$\mathrm{IF}(r_{lf}, r_{hf}) = \begin{cases} \textbf{True} & \text{if outlier} \\ \textbf{False} & \text{if not outlier} \end{cases} \tag{5.25}$$

Isolation forest, by its nature, neglects "direction" in which an outlier is found. This implies that also a PSD estimate which displays a **lack** in energy with respect to the median can be classified as an outlier.
Such a behavior is non-representative of a glitch, and we do not want to classify these realizations as glitch outliers. In order to avoid these misclassifications, the following "excess" condition is implemented:

$$E(r_{lf}, r_{hf}) = \begin{cases} \textbf{False} & \text{if } r_{lf} \leq \bar{r}_{lf} \textbf{ and } r_{hf} \leq \bar{r}_{hf} \\ \textbf{True} & \text{otherwise} \end{cases} \tag{5.26}$$

with $\bar{r}$ representing the median of the feature and is False whenever both r_{lf} and r_{hf} are below their respective ensemble median, True otherwise.

To claim an "glitch detection" a **logical and** ($\wedge$) is then implemented, so that the glitch classifier G is defined as

$$G(r_{lf}, r_{hf}) = \mathrm{IF}(r_{lf}, r_{hf}) \wedge E(r_{lf}, r_{hf}) \tag{5.27}$$

A scatter plot in Figure 5.9 shows the range of variability of the two features before the glitch rejection is implemented (on the left) and after its implementation (on the right) on O3 data. On the left hand side we can see that most of the estimates are found to lay close to zero, while there are some very evident outliers that display a very large value for both features. Another outlier is found close to the central cluster of inliers, which may (or may-not) be a real glitch.
On the right hand side, after the glitched estimates have been substituted, one sees that most values are close to each other, so that none of the estimates should now be contaminated by the glitches' presence.

Discarding the outliers, correlation with cWB

To check whether this procedure actually "heals" our data, one should compare the results against some robust pipeline.
A good baseline can be the output of the standard cWB whitening method, which has provided good results so far. If the two whitening procedures where identical, we would expect a scatter plot of the output of the data whitened with the two methods to coincide with the bisector. For two different whitening methods, instead, we expect a scatter plot of the two outputs to be scattered around the bisector, with only small deviation from it and a high correlation between the two outputs. Therefore, to exclude pathological effects of the new procedure, we compare the standard pycWB whitening and the MESA whitening by checking their correlation and if they align along the bisector. Each point represents the value of the whitened time series at a specific time, with coordinates corresponding to the values obtained from the two different methods. For MESA, we show the outputs of both the whitening procedure with and without the glitch rejection, and for the autoregressive orders 500 and 800.
The results are displayed in Figure 5.10 with a scatter plot of the output time series of the various whitening methods: standard pycWB, as a reference, MESA ar 800, both with and without outlier rejection and MESA ar 500 with the outlier rejection. The data are those relative to the 9th job of the 2nd chunk of O3, for the L1 detector in the GPS time interval ranging from 1238807939s to 1238809139s. As one can see, the correlation plot for cWB against the MESA ar 800 with **no rejection** for the outliers (bottom left) has two different components: the first component is more populated and shows a high correlation with most of the points aligned around the bisector. The second component is populated by the points which drift apart from the diagonal. These points are those whose whitening estimates are affected by the glitch presence, and their variability along the "MESA axis" is smaller than the one along the "cWB axis". The range of variability is a different representation of the standard deviation, hence a smaller variability along the MESA axis means that the standard deviation of the whitened data is smaller than the expected value of 1, compatible with what we have shown in Figure 5.7: It is interesting to notice that the total correlation between the two time series is still pretty high, with a correlation value of $c = 98.6\%$. The relatively high value for this coefficient can be explained considering that most of data belong to segments which are not contaminated by glitches.
On the bottom center, the plot shows the correlation between cWB and MESA ar 800 with outlier rejection. The correlation doesn't show a very large change, being $c = 98.8\%$, but

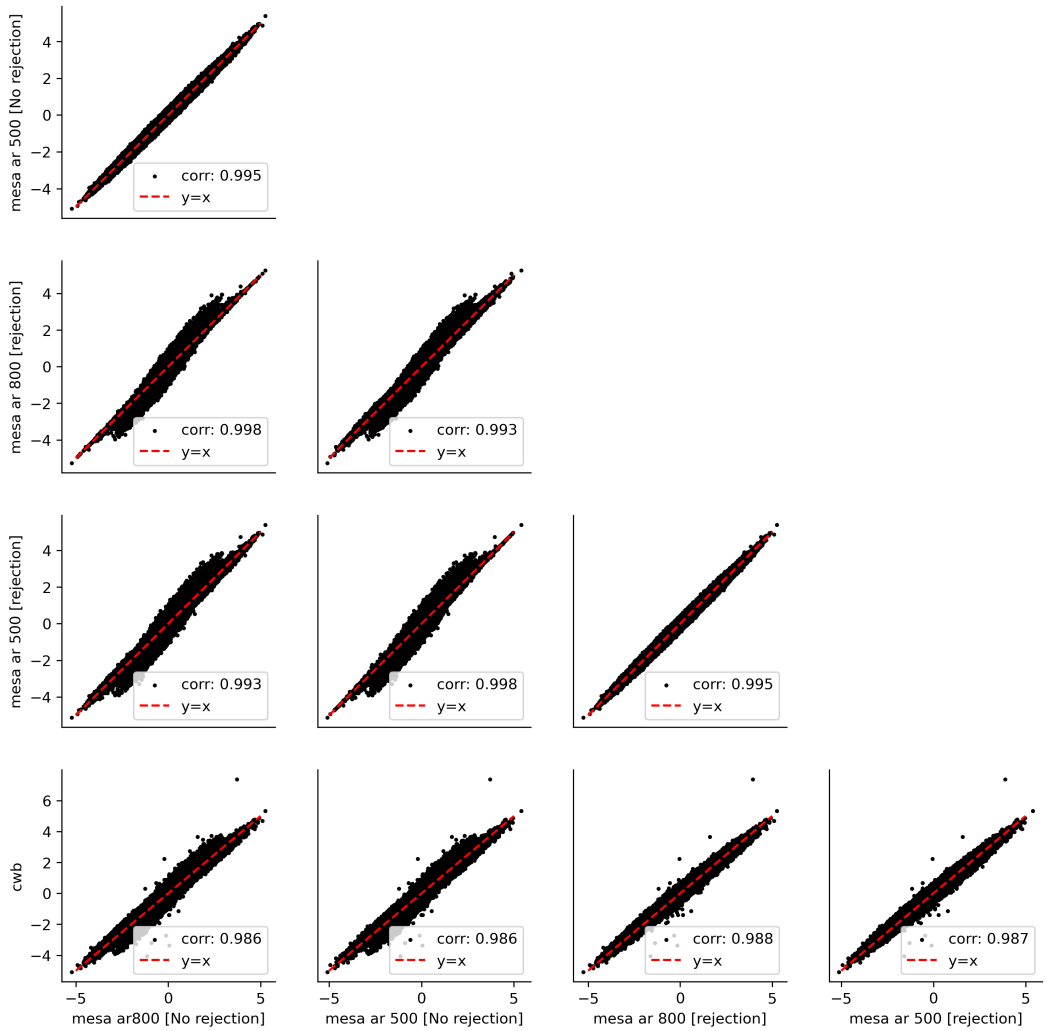


Figure 5.10: Scatter plot of the output of the whitening filter (i.e. the whitened data) with and without Isolation Forest glitch rejection against the cWB whitening. The two methods are highly correlated, with a correlation coefficient of 0.98 for both autoregressive orders

almost all points are now aligned around the diagonal, meaning that outlier rejection has been successful in removing the glitch contamination. Further investigations are provided in the next sections to prove the safety of the procedure.

5.5 Comparison between MESA and pycWB whitening

Having introduced the theory, developing a new whitening algorithm based on MESA and discussing the tuning of the WDM transform, our next step is to compare the two methods and see if these modification provide any advantage with respect to the standard analysis. So far, we have not discussed the problem of selecting the autoregressive order ar for MESA. In this chapter, we now address both tasks simultaneously: comparing the two whitening methods and evaluating their performance with different parameter choices. For pycWB, we consider two values of β-order of the WDM: $\beta = 6$ (current value used by cWB in LVK analysis) and $\beta = 2$ (closer to the Gabor Limit). For MESA, since at this stage it is independent from the WDM transform, we only have to select the autoregressive order ar. We consider two distinct values for the autoregressive order: $ar = 500$ and $ar = 800$. Previous analyses demonstrated that excessively large orders yield diminishing returns, substantially increasing computational cost for marginal improvements in the PSD estimate. While a baseline of $ar \sim 1000$ was suggested in [88], empirical testing around this value established $ar = 800$ as an optimal trade-off between computational efficiency and estimation quality. However, since spectral leakage scales with the autoregressive order, we also evaluate the lower value of $ar = 500$ to minimize this effect. We perform three different studies.
The first investigation plans to compare the normality of the whitened data, together with the compatibility of the mean and standard deviation of the output data with the expected values of 0 and 1 respectively. The data, divided in chunk, are whitened with the two different methods with their respective tunings. To compare the various methods, the data on the chunks of O4a are be used. An Anderson test is performed to assess their normality and the mean and standard deviation of the whitened data are computed for every sub segment.
The second comparison compares the whitened signal against the "true" whitened signal. Several signals are injected in simulated noise, generated to be representative of some specific PSD. Having an a priori knowledge of the PSD allows to compute the "true" whitened signal by using the theoretical PSD to whiten the signal. This allows to make a one to one comparison of the whitened signal with the true whitened signal with some ad hoc statistics.
The third comparison focuses on the effect of the whitening on the leakage of the signal. Again, we look at O4 data, used to build the whitening filters, and the signals are studied in zero noise, so that the actual "end" of the signal is known and the leakage can be easily quantified.

The analysis is performed on the O4a dataset, which is the first part of the O4 observation run, spanning from May 24, 2023, to January 16, 2024. The sampling rate for the analysis is 2048 Hz[4], and the analyzed frequency range goes from 16Hz to 1024Hz. To compare the performance of the various whitening methods, for the WDM transform we adopt the finest available frequency resolution of $\Delta f = 1Hz$.

5.5.1 White noise

The first step in the comparison concerns the analysis of the output of the whitening filters applied to O4a data [38, 45].

In this study, we consider the results obtained with both $\beta = 2$ and $\beta = 6$ for pycWB, and with autoregressive orders $ar = 500$ and $ar = 800$ for MESA. To assess normality, we employ the Anderson–Darling test [22], a widely used statistical test in the context of GW data

[4]The downsampling is preformed in the time-frequency domain, by selecting the relevant frequency bins and discarding those whose frequency is larger that the relevant Nyquist frequency

analysis. The null hypothesis of the test is that the data follow a normal distribution, and the test statistic is computed by comparing the empirical cumulative distribution function of the data with that of the corresponding normal distribution. To perform the analysis, we use the Anderson–Darling test implemented in SciPy [55]. This implementation automatically rescales the data to have zero mean and unit variance. As a consequence, deviations from the expected values $\mu = 0$ and $\sigma = 1$ cannot be detected. For this reason, the mean and standard deviation of the whitened data is studied separately to the normality test. Furthermore, this implementation only returns the value of the test statistic without a p-value. Hence, to compute the p-values, we construct the null distribution of the test statistic by simulating 10^7 realizations of pure white noise with zero mean and unit standard deviation. The p-value is then obtained as the fraction of null-distribution statistics that exceed the observed statistic.
In this section, we focus on datasets whose behavior is not anomalous. Significant deviations from the null hypothesis can occur due to non-stationarities in the data, such as glitches or other transient noise artifacts. Although such deviations are certainly of interest for other studies, they are beyond the scope of this section, which is specifically aimed at evaluating the quality of the whitening procedure under "good" data conditions. Extreme outliers, being rare and sparse, are not included in the analysis. We define outliers as either: p-values that are exactly equal to zero (which represent approximately 3% of the data), or points that fall far (outside 5σ) from the bulk of the distribution. These outliers are sparse and are mostly compatible with data segments contaminated by loud glitches. They are excluded from the following analysis, but the interested reader can find a short discussion on the outliers in Appendix D, together with the Q-transform of some of the relevant outliers.
For every job segments, the mean, standard deviation and p-values are computed consistently with the slicing procedure of the MESA whitening: at every step the PSD is computed over a segment of 15 seconds and it's used to whiten the central 5 seconds. All of the above statistics are computed on this 5 central seconds, producing several samples for each job segment. The samples built this way are not independent, since every $i - th$ sub segment shares two thirds of the data with the previous an next segment $(i - 1,\ i + 1)$ and one third with the $i - 2$ and $i + 2$ segments.
For this reason, the results are resampled to break the correlation between the segments and have a more reliable estimate of the statistics.
To make a comparison on the exactly same data, only the jobs completed in all the 4 analysis are considered, and the outliers of one analysis are removed from all the other analysis. In this way, the results are directly comparable, since they are computed on the exact same data.

At first, we focus on the mean and the standard deviation of the whitened data. Figure 5.11 shows the results for the mean of the whitened data, displaying, from left to right $\beta = 2$, $\beta = 6$, $ar = 500$, $ar = 800$.

All the values for the mean are compatible with zero, with pycWB exhibiting a smaller standard deviation than MESA. Changing the parameter value for both pycWB and MESA has a small negligible impact on the distribution of the mean. The data being zero mean should be no surprise, since the input data already have zero mean. For the standard deviation, the results are displayed in Figure 5.12. In this case, we notice that pycWB displays a biased estimate, with a mean value for the standard deviation of $\bar{\sigma} = 0.982 \pm 0.005$ and $\bar{\sigma} = 0.983 \pm 0.005$ for $\beta = 2$ and $\beta = 6$ respectively. The target value of $\sigma = 1$ is not reached within a 3σ confidence interval. This indicates a bias in the estimate of the total noise power in the estimate of the noise energy in pycWB. MESA, by contrast, provides a completely unbiased estimate for the standard deviation, with a mean value of

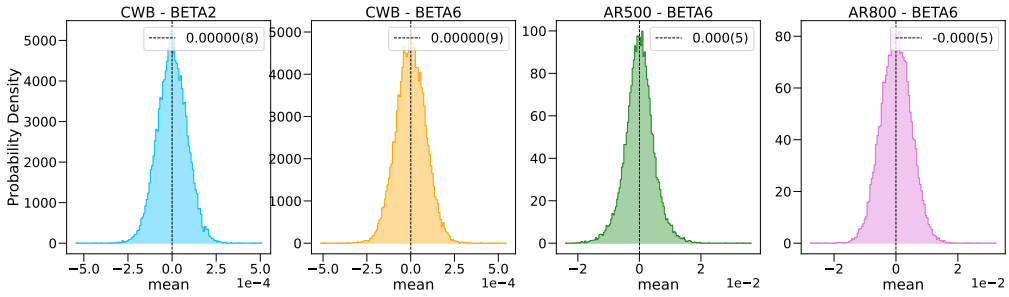


Figure 5.11: The distribution for the mean of the whitened data. From left to right, the results displayed are $\beta = 2$, $\beta = 6$, $ar = 500$, $ar = 800$. The legend report the value for the mean of the plotted distribution, and the error on the last digit is reported inside the parentheses. The mean is compatible with zero for both methods, with pycWB exhibiting a higher precision than MESA.

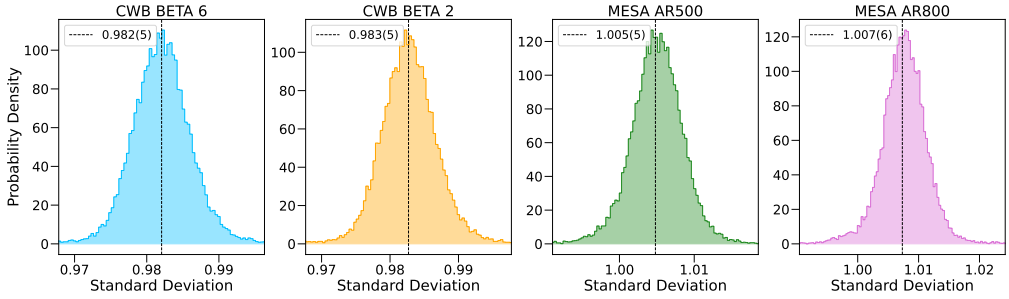


Figure 5.12: The distribution for the standard deviation of the whitened data. The legend report the value for the mean of the plotted distribution, and the error on the last digit is reported inside the parentheses. From left to right, the results displayed are $\beta = 2$, $\beta = 6$, $ar = 500$, $ar = 800$

$\bar{\sigma} = 1.000 \pm 0.005$, compatible with the theoretical value. This demonstrates that the noise power estimate provided by MESA is highly accurate. Again, these results appear to be independent of the tuning parameters for both methods.
While the mean and standard deviation provide some insight, it is plausible that their effect on the final output of the pipeline is neglibile. The most important results are those related to the p-values from the Anderson-Darling test, displayed in Figure 5.13. To compare the results against the null hypothesis, we decided to show the p-values distribution in two different ways in Figure 5.13 for the 10th chunk of the O4 data. On the first, the data are represented as a "PP-plot". The PP-plot shows on the x-axis the cumulative distribution of the number of experiments with uniform spacing. On the y-axis the pvalues are displayed, sorted in ascending order. Since the pvalues are uniformly distributed between 0 and 1, the points should be uniformly distributed along the diagonal line. For visual clarity, since the number of points is very large (10000 independent experiments), we subtract the expected value of the pvalues, so that the smaller fluctuations are not compressed by the larger variability scale going from 0 to 1. On the right side of we combined the p-values using the

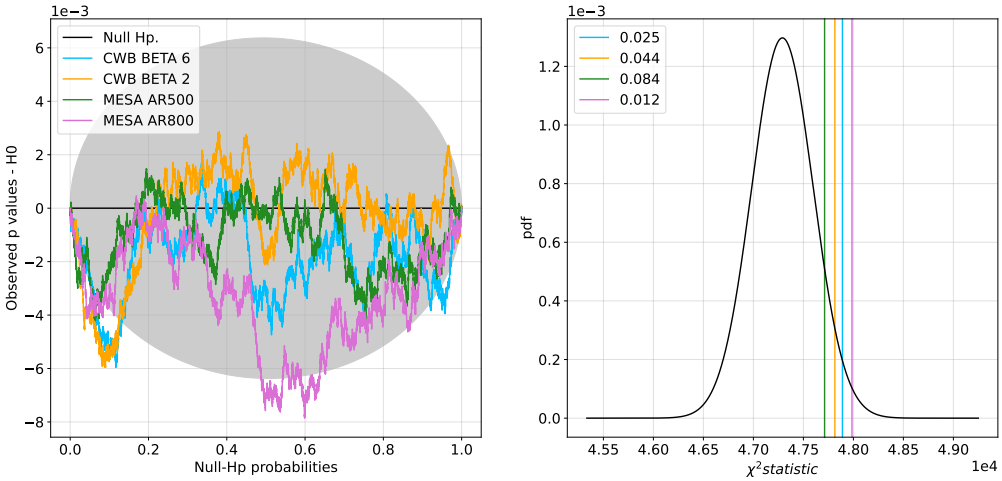


Figure 5.13: The p-values distribution for the Anderson-Darling test on the whitened data. The left panel shows a PP-plot of the p-values for each whitening method and tuning, againt the 90% Confidence Region expected under the null Hypothesis. The right panel shows the p-values combined using the Fisher Formula, with the expected distribution under the null hypothesis. The pvalues reported correspond to the whole 10th chunk of the O4a data

Fisher Formula:

$$\chi^2 = -2\sum_{i=1}^{N}\log(p_i) \tag{5.28}$$

with N the number of independent pvalues and χ^2 being asymptotically distributed as a χ^2 with $2N$ degrees of freedom under the null hypothesis.
In the PP-Plot, the black, horizontal line represent the expected values for the p-values under the null hypothesis, while the gray shaded area represents the 95% Confidence interval around the null hypothesis. Both the plots do not show significant differences between the various method, with oscillations that do not seem to depend on the tuning of the parameters. At the same time, all the methods show a significant excess of low p-values, which is an indication of a significant deviation from the null hypothesis. This is not surprising, since the data are not stationary and with this systematic test we are probably stressing way too much the normality assumption.

The pvalues for all the methods and all the chunks (with exception of Chunk 1 due to some problem in importing the data) are reported in Table 5.1. From the table we can notice that only few pvalues are compatible with the null hypothesis: This is well understood, computing the pvalues we are stressing the assumptions that data are stationary and normally distributed, while this is only an approximation that can fail for different reasons.
From table 5.1 we notice that for MESA the p values are the highest in 50% of the cases. The two cWB cases report less instances of high p-values, mostly compatible between them. Instead, MESA with AR = 800 is never found to provide a p value larger than those obtained with the other methods.

Table 5.1: P-Values from the Anderson-Darling test for the whitened data. The p values is computed using the Fisher Formula over the whole set of pvalues

	pycWB BETA2	pycWB BETA6	AR800	AR500
K02	1.13e-03	2.00e-02	3.25e-05	**5.07e-01**
K03	5.20e-02	**1.14e-01**	1.46e-02	8.43e-04
K04	**1.86e-01**	2.34e-02	4.92e-03	8.79e-03
K05	3.80e-03	2.38e-04	5.43e-03	**1.09e-02**
K06	1.22e-03	**2.68e-02**	9.64e-05	4.02e-04
K07	1.56e-03	1.02e-02	2.18e-02	**1.75e-01**
K08	2.91e-02	5.78e-01	4.77e-02	**7.89e-01**
K09	**9.47e-02**	1.07e-02	3.53e-04	7.48e-03
K10	4.50e-02	2.62e-02	1.23e-02	**8.64e-02**
K11	**5.45e-04**	3.88e-09	3.47e-08	1.88e-08
K12	1.48e-04	5.41e-03	2.06e-03	**2.76e-01**
K13	**6.45e-03**	4.33e-06	3.20e-03	4.56e-03
K14	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$
K15	1.69e-05	3.73e-05	6.86e-06	**5.79e-04**
K16	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$	$< 10^{-16}$

5.5.2 Signals

To study the impact of whitening filters on the signal, zero noise injections are a good way to understand some general properties of the whitening procedure. The reason to study synthetic, stationary data is twofold: the synthetic data provide us a benchmark against which the estimator can be compare, making statements on possible systematics effects of the estimator. On the other hand, stationarity allow us to have several replicates of the very same process, assessing the variability of the estimator. This is the first part of the chapter, in which we exploit known signals in simulated, stationary noise to compare the quality of the whitening procedures by looking at the modifications induced to the signals. We then come back to simulated signals in real data to study the behavior of the time leakage in the different scenarios.

Overlap: how well is PSD reconstructed

One important property to study the whitening procedure is understanding how well the PSD is reconstructed. In fact, the accuracy in the estimate of the PSD is what matters to implement an effective whitening procedure. In order to put some constraints and make a comparison between the two whitening methods, in this section we work with synthetic data, where the PSD is known a priori and the noise is generated with the `memspectrum.GenerateTimeSeries` package. This allows generating timeseries in both time and frequency domains with a given PSD and duration.
The noise is used as a side-reference to compute the whitening filter, but the signals are injected in a zero-noise environment, so that the effect of the noise on the signals is visible and we can make comparisons directly on the signals.
The signals are generated with `pycbc` [93] functions, as a zero-spin binary system of masses $m_1 = m_2 = 30M_\odot$ at a distance of $d = 400Mpc$ and a right ascension ra = 0, declination dec = 0 and "IMRPhenomXPHM" approximant. A reference PSD is estimated on O3 data,

and a synthetic time series is generated to be correctly representative of the estimated PSD. The signal is then injected in the zero-noise data and whitened using the pycWB and MESA estimated of the PSD, obtaining two different whitened estiamtes h_{cwb} and h_{mesa}. Since the real PSD is known a priori, one can compute the "True" whitening filter and use it to whiten the the signal, obtaining the reference signal in the whitened domain h_{ref}.
As a measure to assess the quality of the PSD estimate, we use an indirect quantity most commonly used in the context of waveform reconstruction, the **overlap** [105].
The overlap is defined as the normalized scalar product, i.e. the cosine, between the reference signal and the reconstructed signal, i.e.,

$$ovlp(h_1, h_2) = \frac{(h_1|h_2)}{\sqrt{(h_1|h_1)(h_2|h_2)}}; \qquad -1 \leq ovlp(h_1, h_2) \leq 1 \tag{5.29}$$

and the scalar product is defined as

$$(h_1|h_2) = \int_{t_0}^{t_1} dt h_1(t) h_2(t) \tag{5.30}$$

with t_0 and t_1 the start and end time of the signal. The overlap is a measure of the similarity between two signals, with a value of 1 indicating perfect agreement and a value of 0 indicating no correlation. To see which method provides a better estimate of the PSD in this case, we want to compare the overlap $ovlp(h_{cwb}, h_{ref})$ with $ovlp(h_{mesa}, h_{ref})$, with h_{ref} the reference signal whitened with the true PSD known a priori, h_{cwb} the signal whitened with pycWB and h_{mesa} the signal whitened with MESA. Since the injected waveform h is fixed, any difference between the reference and the method's whitened signals can only originate from the estimated PSD, $S_n(f)$.

$$\tilde{h}_{\text{white}}(f) = \frac{\tilde{h}(f)}{\sqrt{S_n(f)}}, \tag{5.31}$$

hence, higher values of the overlap can only be explained with more accurate PSD estimate. It is worth mentioning that pycWB and MESA use two different window functions to move across the time and the frequency domain. However, the length of the windows is chosen to be extremely larger than the duration of the signal. Moreover, only the central seconds where the window is flat are retained, so that the differences that may arise from the different windowing are negligible. Figure 5.14 shows the distribution of the overlap for the two methods, pycWB and MESA, with different parameters. The results show that MESA provides a better estimate of the PSD than pycWB, especially for ar = 800. The distribution of the overlap is very narrow, with a very high value for the overlap and no value below $ovlp = 0.996$. This is expected given the simplified scenario: stationary noise and no signal that could affect the PSD estimate. Larger differences and greater variability are expected in real data, where non-stationarities are present; these are discussed in the next chapter.
Overall, MESA provides systematically larger overlap values, indicating better PSD estimation accuracy regardless of parameter choice. For pycWB, results are slightly better for $\beta = 2$ than for $\beta = 6$, though their mean values are compatible within 1 standard deviation. Similarly, MESA results for the two ar values are compatible within 1 standard deviation, with $ar = 800$ providing a marginally better PSD estimate than $ar = 500$.

Leakage

As we have seen in Section 5.3.1, the whitening procedure comes with a time leakage of the signals dependent on the non diagonality of the whitening filter. In this section, we come back

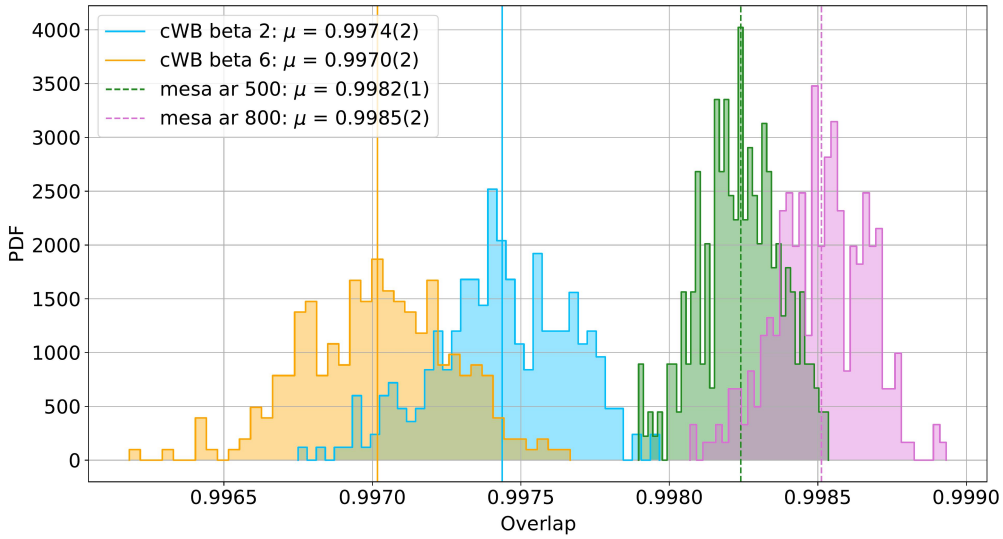


Figure 5.14: The distribution of the overlap for the two methods, pycWB and MESA, with different parameters. The results are displayed for $\beta = 2$ and $\beta = 6$ for pycWB, and $ar = 500$ and $ar = 800$ for MESA. The results show that MESA provides a better estimate of the PSD than pycWB, especially for ar = 800.

to this topic, comparing the leakage provided by the two different method in a quantitative way. We perform the same injection again and again in different noise realization, still coming from the tenth chunk of O4 data. To increase the variability of the results, we fix the right ascensions of the injected signal so that, as the earth rotates, the SNR of the injected signal changes in time, ranging from 5 up to 80. This variability allow us to study the leakage as a function the SNR of the injected signal. Due to the linearity of the whitening filter, the leaked SNR scales proportionally with the waveform hence, since it is invariant to waveform scaling[5], we study the ratio of leaked SNR to total SNR at a given time t For CBC signals, the maximum amplitude typically occurs at the merger time, making the post-merger region of the data the most affected by leakage. To quantify this, the leaked SNR is computed within a post-merger window starting at t^*, the time at which the signal ends, and extending up to $t_{end} = 5$ seconds. To reduce the statistical fluctuations in the leaked SNR estimate, the post merger window is divided in time-bins of 50 ms from t^* to $t = 2$ seconds, and bins of 100 ms duration from $t = 2$ seconds to t_{end}. These values have been chosen after some attempts to balance the need for a sufficient number of samples in each bin to reduce statistical fluctuations, while also ensuring that the bins are small enough to capture the time evolution of the leakage. The change in size of the bin is motivated by the interest in the leakage behaviour close to the merger, where the leakage is expected to be more pronounced, while is less relevant at later times, since its overall contribution is found to be small ($O(10^{-3})$). The leaked SNR is calculated for each bin as an average of the SNRs in that bin, i.e.

$$SNR_L(\bar{t}) = \frac{1}{N_{bin} SNR_{signal}} \sqrt{\sum_{t_i \in bin} h_w(t_i)^2}. \tag{5.32}$$

[5]This is only True for a fixed waveform morphology. Nevertheless, since most of the leakage is caused by the maximum amplitude of the waveform, the ratio remains a good statistic for our scopes

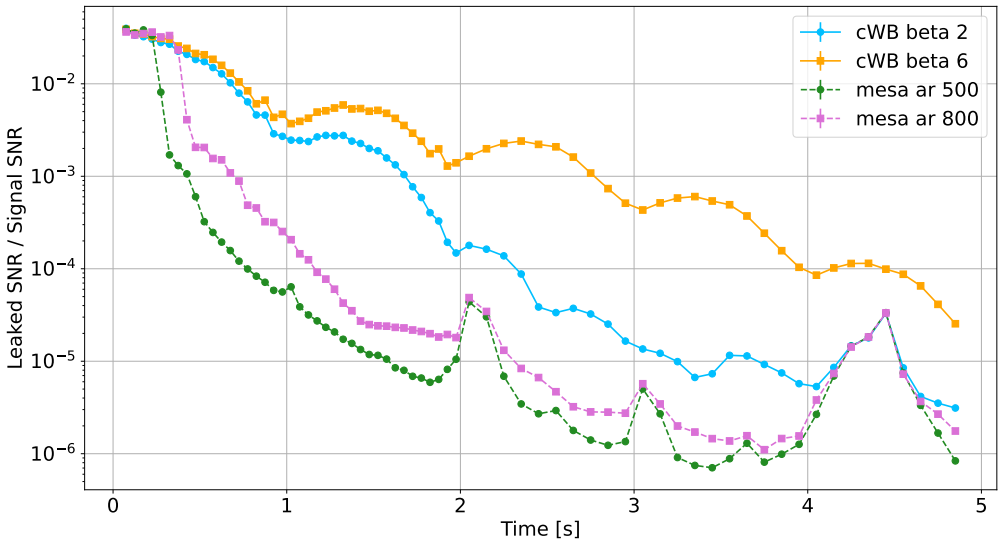


Figure 5.15: The distribution of the leakage for the two methods, pycWB and MESA, with different parameters. The results are displayed for $\beta = 2$ and $\beta = 6$ for pycWB, and $ar = 500$ and $ar = 800$ for MESA. The results show that MESA provides a better estimate of the PSD than pycWB, especially for ar = 800.

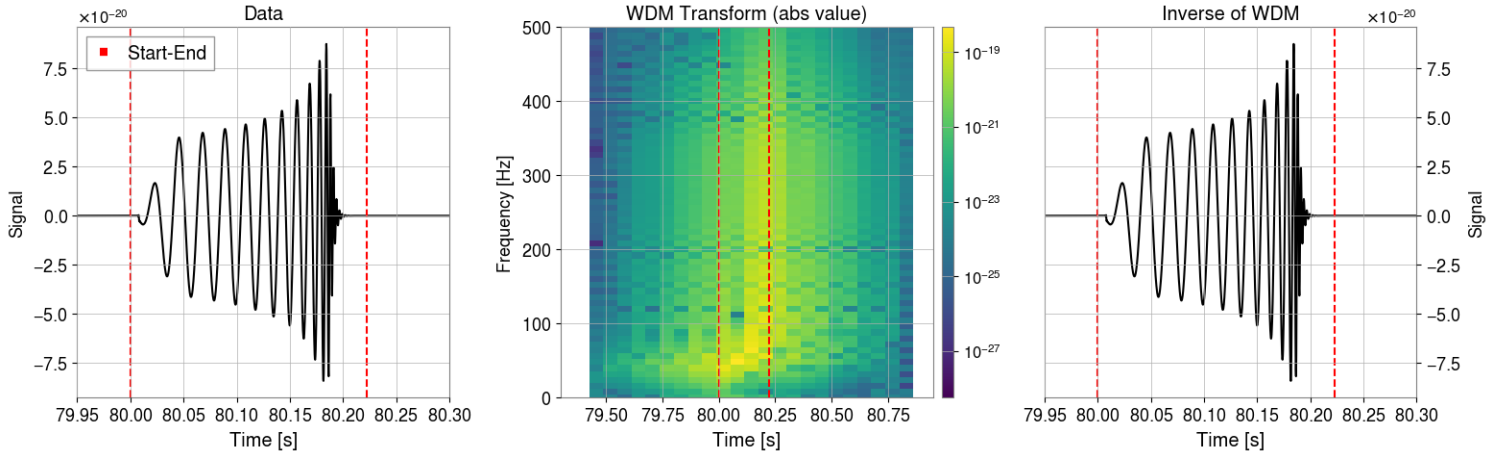
with $\bar{t}$ being the central time of the bin and N_{bin} the number of time steps in the bin and $h_w^{(t_i)}$ the whitened signal.
The results obtained with the different methods are shown in Figure 5.15. During the first $\sim$ 150 ms, there is almost no difference between the two methods, with only a weak dependence on the corresponding parameters. Beyond this point, the behavior of MESA and pycWB diverges significantly:
The leaked SNR for MESA-ar500 begins to drop exponentially, which can be attributed to the length of the autoregressive filter, as illustrated in Figure 5.6. Since most of the filter's energy is concentrated within the first "ar" seconds, the leakage is expected to decrease at approximately $500/2048 \approx 0.244$ seconds. A similar behavior is observed for MESA-ar800, where the leakage drops at roughly $800/2048 \approx 0.390$ seconds. After this initial drop, both curves exhibit a comparable trend, with $ar = 800$ showing a slight offset lasting up to about 4 seconds. This offset is accompanied by some complex structures related to the noise properties and the temporal shape of the whitening filter.
The behavior of pycWB is markedly different, with the leakage decreasing much more slowly and reducing by one order of magnitude after 800 milliseconds. The filter is then dominated by an oscillatory systematic effect. This systematic is related to the time-domain shape of the WDM base function, with the oscillations in the leakage being consistent with the oscillations of the lobes of the square of the WDM base function in the time domain. The oscillations are more pronounced for $\beta = 6$ compared to $\beta = 2$, with the latter eventually converging to MESA's behavior after approximately 4 seconds, while the former remains dominated by the systematic effect.
This is an interesting behavior, and is a clear indication that the leakage related to the whitening filter is not the only source of leakage in the pycWB pipeline. There is a second, dominant source of leakage relate to the pixel representation in time frequency domain.

Leakage from the Wavelet Transform

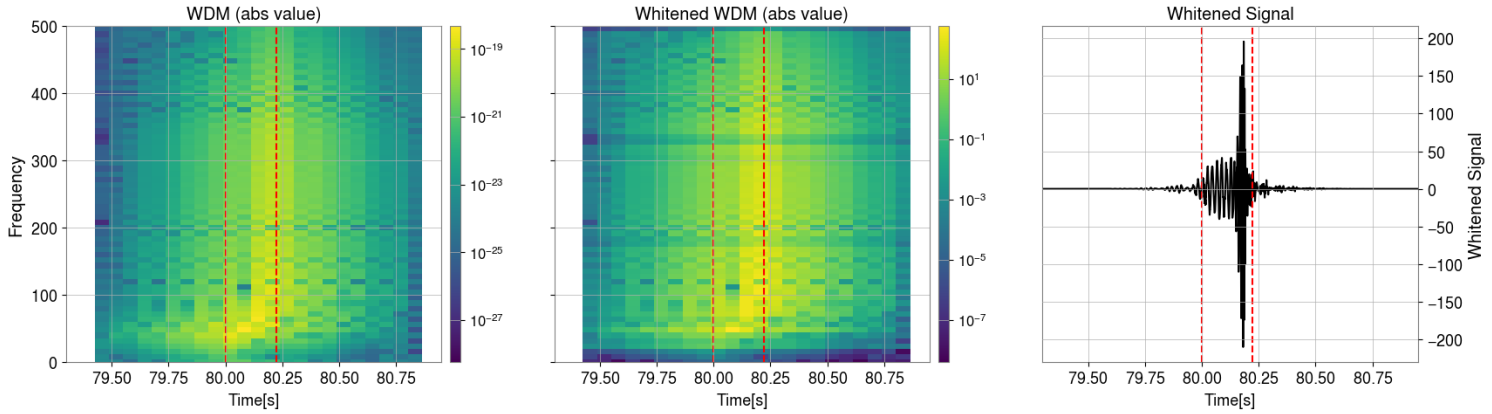
As we have shown in Figure 5.1, the WDM base function in both time and frequency domain are not a delta function, but spread over both the time and frequency domain. When building the pixel, we associate to the pixel the time at which the maximum of the wavelet envelop is reached. Nevertheless, the neighboring data points contribute to the pixel value, and they are weighted less and less the further we move from the time t_i. Each pixel is then "polluted" by the information contained in the neighboring data. The larger the envelope of the carrier wavelet, the further information can leak in the time frequency representation. Since the WDM was built to have a good frequency resolution, for the Gabor principle, it must have a less precise envelop in the time domain. This would not be a problem if we were considering stationary noise only, since the effect would average out across different pixels. Also, since the transform is invertible, we only expect the leakage to be present in the time frequency representation, but to disappear when going back to the pure time domain.

We show this effect in Figure 5.16a, where we plot the original signal in time (left), its WDM representation (center), and the the time series obtained by inverting the WDM representation (right). In the time domain the signal is only present between the two dashed vertical lines, while it is zero on the outside. When moving to the TF representation, the signal energy leaks outside that region, due to the non-compact representation of the WDM transform. The CBC signal then undergoes the inverse transform (right panel) and the final time series is identical to the original one within numerical precision. This happens because the WDM transform is invertible, and the information is not lost in the TF representation. Different pixels are originated in such a way that the information is preserved, and that their interference ensures that the leakage disappears when moving back to the time domain.
This interference effect is however broken when we apply the whitening procedure, since we are modifying the representation of the data in the time frequency domain. Figure 5.16b shows the WDM representation of the signal before and after the whitening procedure. The left panel shows the WDM representation of the signal, while the right panel shows the WDM representation of the whitened signal. The whitening procedure modifies the representation of the data in the time frequency domain, and it changes the relative weight of the different pixels. This is a problem, since we are modifying the representation of the data in such a way that the interference effect is destroyed. When we divide by the amplitude spectral density, we are modifying the representation of the data in the time frequency domain, and we are changing the relative weight of the different pixels. This means that, at times outside the signal region, the pixels affected by the signal leakage are not properly canceled by the interference with the other pixels, and the signal remains "trapped" in the those regions domain. This is shown but the right plot of the same figure, where the resulting whitened signal is shown and the signal leakage outside the signal region is visible. The effect is less present the more compact the WDM base function is in time, since the signal is less spread over time in the TF representation, explaining why the leakage decreases faster for $\beta = 2$ than for $\beta = 6$, since the former has a more compact envelope in time, and the signal is less spread in time. This also explains the oscillatory behavior of the leakage, since the oscillations are consistent with the oscillations of the lobes of the square of the WDM base function in the time domain.
Therefore, what we are seeing in figure 5.15 are two different effects. On the one hand, for standard pycWB procedure, we are dominated by the leakage from the wavelet transform, while the leakage related to mesa is purely related to the whitening procedure. Nevertheless, for large enough time (4 seconds after the signal) we can see the $\beta = 2$ converges to the MESA behavior, since we moved enough far away from the signal For pycWB, the wavelet

representation is the dominant source of long-time leakage, especially for $\beta = 6$. For $\beta = 2$, this effect is smaller because the time envelope is more compact. At late times, once the wavelet contribution fades, the leakage approaches the transfer-function behavior seen with MESA.



(a) Invertibility of the WDM transform. The left panel shows the time series of a CBC signal, the middle panel shows the WDM representation of the signal, and the right panel shows the time series obtained by inverting the WDM representation. The final time series is identical to the original one, demonstrating that the WDM transform is invertible and that information is not lost in the time-frequency representation.



(b) WDM representation of the signal before and after the whitening procedure. In all the panels the vertical lines show the start and the end time of the original signal. The left panel shows the WDM representation of the signal, the center panel shows the WDM representation of the whitened signal, and the right panel shows the signal in time domain after the whitening procedure. The whitening procedure changes the relative weight of the different pixels in such a way that their combination is non-zero outside the signal region.

Figure 5.16: WDM time-frequency representation: invertibility and effect of whitening.

5.6 Conclusions

This chapter addressed two central aspects of the pycWB pipeline: the tuning of the WDM wavelet transform and the implementation of a novel MESA-based whitening procedure.
Regarding the WDM tuning, we demonstrated that the standard $\beta = 6$ parameter is sub-optimal from a time-frequency localization perspective. By reducing β to 2, we moved the WDM transform approximately 20% closer to the Gabor Uncertainty limit, resulting in a more compact representation of signals in the time-frequency plane. While this tuning does not fundamentally alter the whitening procedure itself, it sets the stage for improved overall analysis performance.
We provided a detailed theoretical analysis of two distinct approaches: autoregressive (AR) and PSD whitening, both implementable via MESA. The key theoretical finding is that these methods are related by a simple but consequential relation in Fourier space: the PSD filter has the same gain as the AR filter but discards all phase information. Critically, the PSD filter preserves signal morphology and phase content, making it the preferred choice for reconstruction-oriented analyses. Both filters exhibit time leakage due to their non-diagonal nature in the time domain, but with different characteristics: AR leakage is causal (one-sided), while PSD leakage is symmetric.
From a practical implementation standpoint, we demonstrated how to construct a robust MESA-based whitening procedure that operates on windowed data segments and incorporates glitch rejection via Isolation Forest algorithms. In the cases studied here, MESA and standard methods give very comparable results in most whitening diagnostics. MESA appears slightly better in some cases, but the difference is small. In particular, the improvement is not dramatic, and the two approaches remain largely consistent.
The analysis of signal leakage, however, shows a clearer advantage. At early times (first ~ 150 ms), both methods perform similarly. Beyond this, MESA leakage decays exponentially at a timescale determined by the autoregressive order, while pycWB leakage persists with oscillatory structure up to several seconds. Crucially, we identified that the dominant long-time leakage in pycWB originates not from the whitening filter itself, but from the WDM time-frequency pixelization linking wavelet compactness to time leakage. Reducing β from 6 to 2 mitigates this effect, providing additional justification for the improved wavelet tuning.
In summary, the developments presented in this chapter—improved WDM tuning and MESA-based whitening—address complementary aspects of noise handling and signal fidelity. The whitening results are broadly consistent across methods, with only mild differences in favour of MESA. The leakage analysis is instead more favorable to the MESA approach.

Chapter 6

Waveform reconstruction in pycWB

6.1 Introduction

Having discussed the theoretical aspects of the whitening procedure, developed a fully compatible implementation of the new method, and compared the two approaches in terms of whitening performance on HL strain data, we now focus on a specific point of the analysis: waveform reconstruction. Although informative, the previous results only concerned the output of the whitening procedure, i.e. where only focused on the pre-processing of the data. In a realistic analysis, however, the quantity of interest is the output of the whole pipeline: the reconstructed waveform. The goal of this chapter is twofold: In the first section, we investigate the impact of both the new mesa-whitening procedure and the WDM tuning on the waveform reconstruction, providing a systematic comparison with the standard methodology. We perform this comparison on different signal morphologies: a larger variety of CBC signal, whose parameters are chosen to be representative of the CBC parameter space as inferred by O3 catalog [12, 13].
In the second section, instead, we focus on the impact of the Likelihood Regulators in pycWB. These Likelihood regulators are a set of parameters of parameters that were developed in the contest of LVK analysis to improve detection efficiency. We focus specifically on the Γ regulator, and determine its impact on the waveform reconstruction.

6.2 Waveform Reconstruction: Comparing MESA and pycWB

In this chapter we focus on the reconstructed waveform by injecting signals into the data and comparing the pipeline's output with the originally injected signal. The analysis is performed on the 10th chunk of O4, chosen because all methods yield similar p-values for this segment (see Table 5.1), and none display particularly small values. We perform two different types of comparison. The first is an All-Sky comparison, injecting signals from different sky directions and comparing the reconstructed waveform with the injected waveform. The second study focus on the time leakge of the reconstructed waveform.
In the previous chapter, the usage of MESA-whitening was completly decoupled from the

choice of the wavelet tuning. In fact, with MESA, the whitening is performed in the pure frequency domain, and the wavelet transform is avoided. However, when we consider the waveform reconstruction, the wavelet transform is a fundamental part of the analysis, and its parametrization is relevant no matter what whitening method is chosen, since the analysis steps coming after the whitening, such as the clustering and the likelihood computation are performed in the time-frequency domain.
In principle, this could provide a significant advantage for the re-setting of the analysis: reducing β from 6 to 2 moves the transform closer to the Gabor limit, at the cost of a decrease in frequency resolution. Combining MESA with $\beta = 2$ could strike a good balance, exploiting MESA's high frequency resolution together with the greater time–frequency compactness of WDM with $\beta = 2$.
We consider the following combinations of whitening method and wavelet tuning:

- pycWB whitening with $\beta = 6$, which is the standard configuration used in the LVK analysis.
- pycWB whitening with $\beta = 2$, chosen to check the impact of solely the wavelet tuning on the waveform reconstruction
- MESA whitening with $\beta = 2$ and autoregressive order $AR = 500$, chosen to check the impact of both the whitening method and the wavelet tuning on the waveform reconstruction.

6.2.1 All-Sky Comparison

All the comparison of this chapter are performed on "all-sky" data: the injected sky position for the signals is uniformly sampled across all the sky possible directions, and the signals are distributed in time at a fixed interval of 200 seconds from the start of the analyzed data, with a deviation from this value of ± 5 seconds extracted randomly from a uniform distribution $U(-5, 5)$. We inject 3000 different CBC signals to be representative of the population inferred from the O3 run [13], with the reference lower frequency set to the standard $f_{\text{lower}} = 20\,\text{Hz}$ commonly used in CBC analyses [42]. The masses range from $\sim 2\,M_\odot$ to $200\,M_\odot$, and distances range from $50 Mpc$ to 15 Gpc. The spin parameters range from -1 to $+1$, peaking around zero, with 60% of the signals having a spin $\in [-0.25, 0.25]$.
As a metric, we use the overlap compute between the injected and the reconstructed waveform. The analysis focuses on two different signal morphologies: CBC signals and Burst Injections. The specifics of the signals parameters and morphologies are given in their respective sections.
The section ends with a discussion of the leakage of the reconstructed waveform for CBC signals. In Appendix E we report a similar study on two case examples for CBC signals, which is more focused on the ensamble of reconstruction for a specific signal, opening to different types of comparison between the whitening methods but limited to the signals under scrutinity.

CBC signals

The CBC signals are generated using the `pycbc` [93, 115] library, specifically employing the `IMRPhenomD` waveform model [98], The injected parameters for the CBC signals are chosen to cover a wide range of the parameter space and, to remain astrophysically motivated, the distributions of the intrinsic and the distances of the injected signals are taken to be

Table 6.1: Median value of 1 - Overlap between the injected and reconstructed waveform for the various whitening methods.

pyCWB BETA6	**pyCWB BETA2**	**MESA AR500 BETA2**
0.255(7)	0.243(7)	0.244(6)

representative of the astrophysical population of CBC sources as inferred from O3 observations [12, 13].
The results of the analysis are reported in Figure 6.1, where the distribution of the $1 - ovlp$ between the injected and reconstructed waveform is plotted for the various whitening methods. In the boxplot, the box ranges from the first quartile (Q1) to the third quartile (Q3) of the data, with a line at the median (Q2). The "whiskers" extend from the 5th percentile to the 95th percentile of the data, providing a visual representation of the spread and variability.

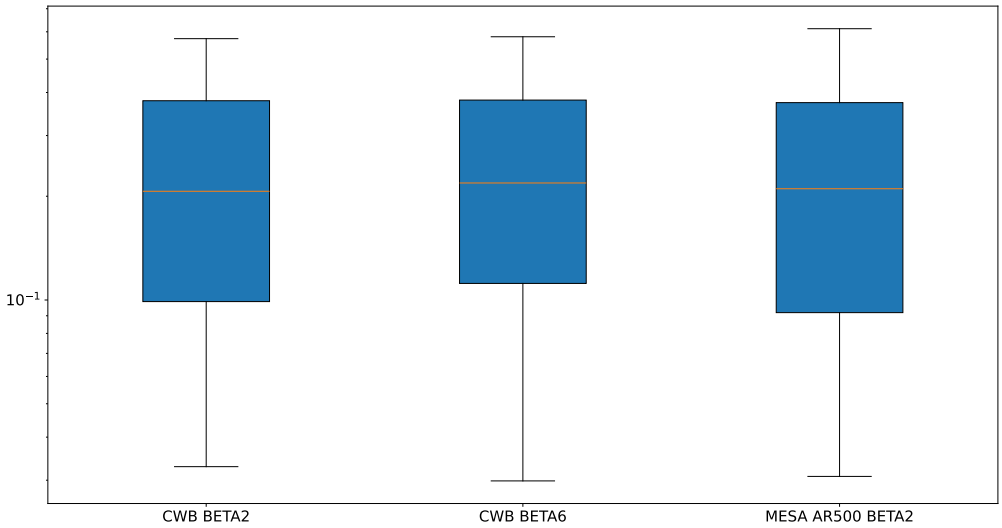


Figure 6.1: Boxplot of the distribution of $1 - ovlp$ between the injected and reconstructed waveform for the various whitening methods. The results are displayed for cWB with $\beta = 2$ and $\beta = 6$, and MESA with $ar = 500$. The log scale allows better visualization of the differences between the methods, with smaller values being preferred.

From the plot, the various boxplots do not show any significant difference among the four methods. All boxplots display similar values across the plotted percentiles, with only the whitening obtained with MESA at $ar = 800$ showing a slightly better performance. The median values of the overlap for the varios methods are reported in Table 6.1, together with their uncertainty. The median of the distribution of $1 - overlap$ is slightly larger for the standard cWB method is consistent with the other methods within the uncertainty.

Burst Injections

Together with the CBC signals, we consider a set of injections covering specific waveforms morphologies. We refer to these injections as "Burst". These injections are widely used

in burst searches and LVK analysis [38, 5], and are composed by three main classes of waveforms:
Sin-Gaussian (SG) waveforms: sinusoidal oscillations modulated by a Gaussian envelope, defined by central frequency f_0, quality factor Q, and amplitude A.

$$h_+ = Ae^{-\frac{t^2}{\tau^2}} \sin(2\pi f_0 t), \quad h_\times = Ae^{-\frac{t^2}{\tau^2}} \cos(2\pi f_0 t) \tag{6.1}$$

with $\tau = \frac{Q}{\sqrt{2}\pi f_0}$. The quality factor are chosen to be $Q = 3, 9, 100$ and frequencies betwen 70 Hz and 1300 Hz;

Elliptic Sine-Gaussians (SGE): a generalization of SG allowing elliptical polarization

$$h_+ = A\left(\frac{1+\cos^2\epsilon}{2}\right) e^{-\frac{t^2}{\tau^2}} \sin(2\pi f_0 t), \quad h_\times = \cos\epsilon A e^{-\frac{t^2}{\tau^2}} \cos(2\pi f_0 t) \tag{6.2}$$

where ϵ is the ellipticity parameter extracted uniformly in the range $[0, \pi]$. The other parameters are chosen as for the SG waveforms.

White Noise Bursts (WNBs): broadband signals with a flat spectrum within a specified band, characterized by their central frequency f_0, their bandwidth Δf, their duration τ and their amplitude. The WNB are generated with the three following configurations: $f_0 = 150Hz, \Delta f = 100Hz, \tau = 0.1s$, $f_0 = 300Hz, \Delta f = 100Hz, \tau = 0.1s$, $f_0 = 700Hz, \Delta f = 100Hz, \tau = 0.1s$.
The SNR distribution of these waveforms is chosen to be $1/SNR$. The number of injections is of the order of 10^4 so that every waveform is sufficiently represented in the distribution of the overlap.

The results of the analysis over the set of burst waveforms are represented in Figure 6.2 as a box plot as a function of both the type of waveform and the whitening method. The box are defined as in the CBC all-sky analysis.

For every waveform morphology, we display three different boxplots. The green box, at the left side of each waveform group, represents the results for the MESA whitening with β = 2. The central box, in red, represents the standard choice for the parameters in pycWB ($\beta = 6$). The blue box, on the right, represents the results for pycWB with $\beta = 2$.
By Looking at the plot, there is not meaninguful difference between MESA and pycWB with $\beta = 2$, with no systematic improvement of one method over the other. However there is a systematic improvement of both these methods against pycWB with $\beta = 6$. From the plot, it is evident that pycWB with $\beta = 6$ shows the worst performance among the three methods. It achieves systematically larger values (i.e. worst reconstruction) of $1 - ovlp$ for the 5th, 25th and 50th and 75th percentiles of the distribution. The gain on the median is of 1% across all the waveforms. There is no significant difference, instead, when looking at the 95th percentile, where all three methods show comparable behavior within the fluctuations. The rightmost boxplot represents the results obtained marginalizing over the waveform morphology. The marginalized plot is consistent with all the above conclusions, with MESA and pycWB with $\beta = 2$ showing a significant improvement over pycWB with $\beta = 6$ and no significant difference between the two methods with $\beta = 2$.
These results show that there is no significant difference in on the reconstructed waveform between the two whitening methods, while the wavelet tuning has got a measurable impact on the reconstruction. Rather than a rigid translation toward better overlap values (the $1 - ovlp$ 95th percentile is fixed), the improvement of the wavelet tuning is more a widening of the distribution toward better overlap values for the reconstruction.

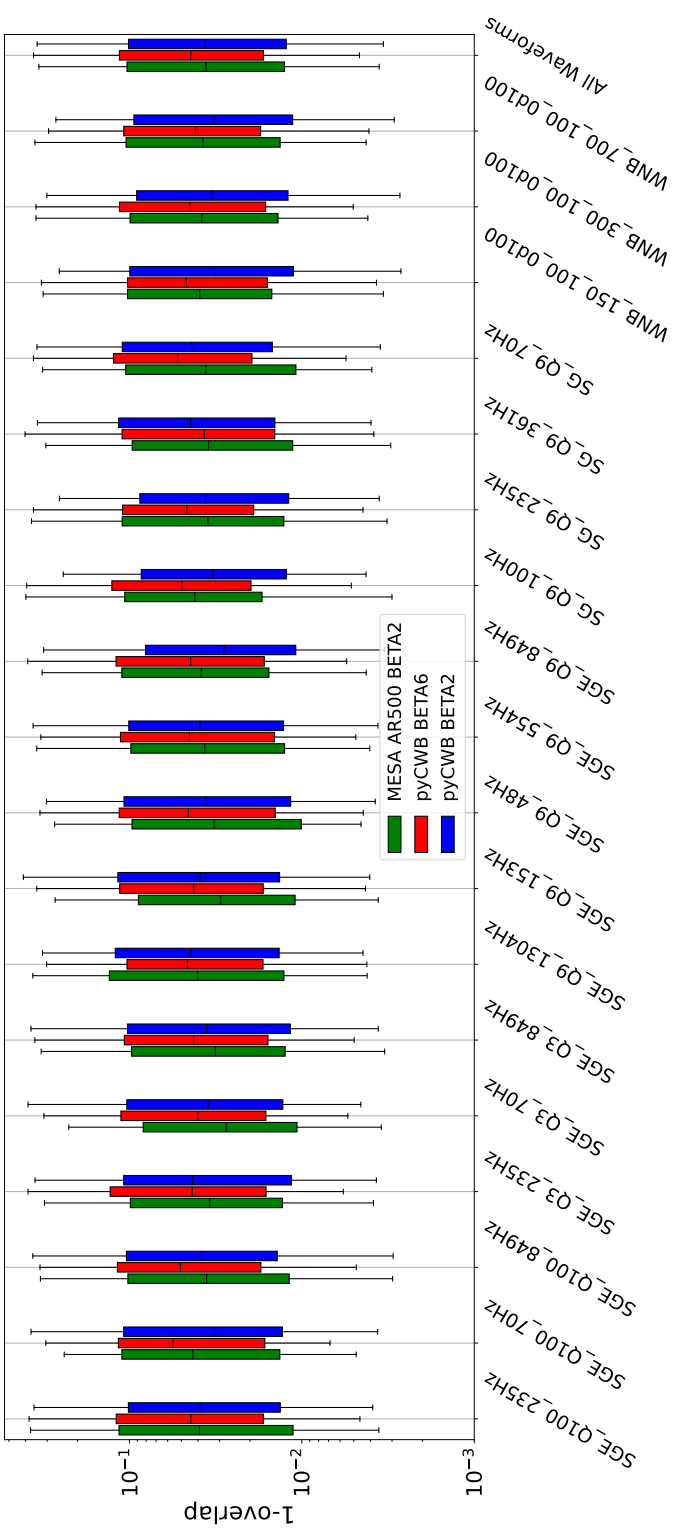


Figure 6.2: Boxplot of the distribution of $1 - ovlp$ between the injected and reconstructed waveform for the various whitening methods for waveforms of generic morphology. The results are displayed for cWB with $\beta = 6$ (red), cWB with $\beta = 2$ (blue), and MESA with $ar = 500$ (green). The whiskers extend from the 5th to the 95th percentile of the data, providing a visual representation of the spread and variability. The edges of the box represent the first and third quartiles, with a line at the median.

6.2.2 The Leakage

Leakage is computed following the same procedure as in Section 5.5.2, evaluating the leaked SNR in the post-merger window from t^*—the signal end time—to $t_{\text{end}} = 1\,\text{s}$. The reference time t^* is defined on the injected strain waveform as the time at which the signal energy has decreased by a factor 10^3 with respect to its peak value. Since the leaked SNR is evaluated locally as a function of time, the result is robust to this specific definition: changing the start of the post-merger window only shifts the time origin from which leakage is computed, without altering the leakage profile at fixed time lag.
The leaked SNR is normalized by the injected SNR, yielding a dimensionless quantity comparable across different injections.
Figure 6.3 shows the leaked SNR as a function of time for for all whitening methods together with the wavelet tuning of the β order. Other than the wavelet parameter, only the resolution levels of the wavelet transform used in the multi-resolution analysis using the following sets of resolution levels:

- $\Delta f_{min} = 1Hz$, $\Delta f_{max} = 64Hz$
- $\Delta f_{min} = 2Hz$, $\Delta f_{max} = 128Hz$
- $\Delta f_{min} = 4Hz$, $\Delta f_{max} = 256Hz$

The results in Figure 6.3 differ markedly from those in Figure 5.15. Prior to waveform reconstruction, leakage showed clear differences between MESA and pycWB, particularly at larger times. Here, instead, the dominant contributions to post-merger leakage follow is more similar across methods and an overall similar behaviour, but with a clear hierarchy among three distinct parameters. The leading-order effect is determined by the choice of resolution levels. During the multi-resolution analysis, pixels are selected based on their energy content, and limiting the maximum resolution level constrains the maximum length of the selected pixels. This reduces both the temporal spread of individual pixels and the leakage from neighboring data. The displacement along the x-axis between curves corresponding to different maximum resolution levels directly reflects this: going from level i to level $i-1$, the time pixel length is halved, and this factor-of-2 scaling is clearly visible in the leakage curves — with an approximately factor-of-4 displacement between the level-8 and level-10 cases. The second-order effect is produced by the choice of the β parameter. For both pycWB and MESA, $\beta = 2$ yields smaller leakage in the post-merger window compared to $\beta = 6$. This follows directly from the pixel representation of the signal in the time–frequency plane: each time pixel is built as an FFT of the time series over a window defined by the WDM basis-function envelope, and a smaller β corresponds to a wavelet more compact in time. A sharper temporal envelope results in a shorter effective window, reducing leakage from neighboring data. In this regime, the dominant source of leakage is therefore the pixel representation of the signal in the time–frequency domain, rather than whitening-induced leakage.

The third-order effect is the choice of whitening method, with MESA providing slightly smaller leakage than pycWB. This is consistent with the results of the previous section; however, the difference between the two methods is subdominant with respect to both the resolution level and β choices, confirming that the WDM pixel representation is the primary driver of post-merger leakage in the reconstructed waveform. This behavior is further illustrated by the time–frequency likelihood maps shown in Figure 6.4 for the three resolution level configurations. Reducing the maximum resolution level produces visibly more compact pixels in time, directly reducing their temporal spread and, consequently, the leakage in the post-merger window.

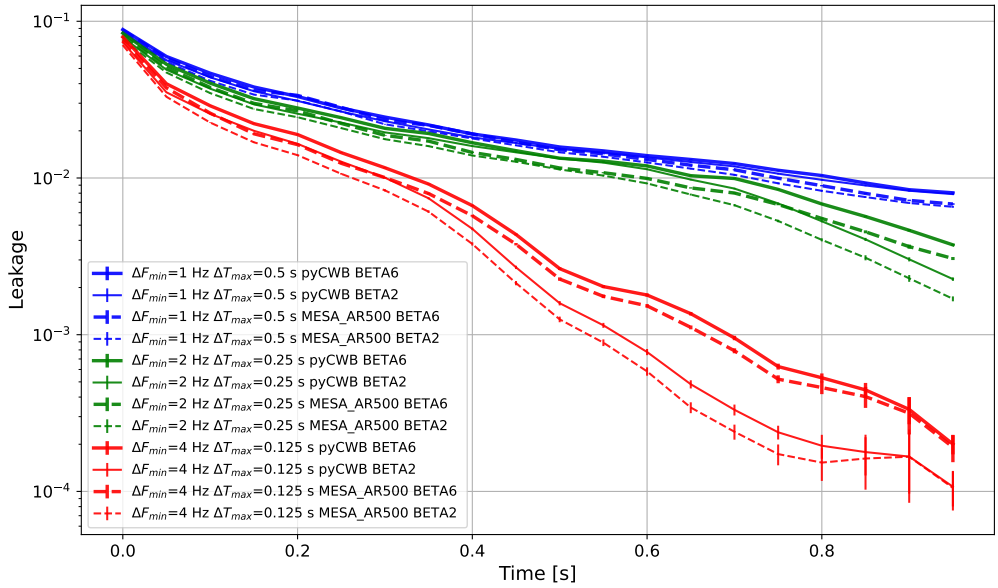


Figure 6.3: The distribution of the leakage for the reconstructed waveform, as a function of the maximum resolution level used in the multi-resolution analysis. The results are displayed for three different sets of resolution levels.

6.3 Likelihood Regulators

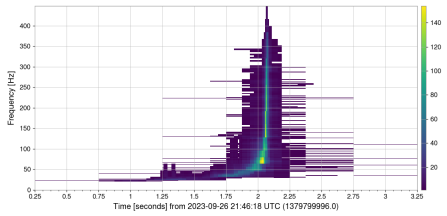
In this section we study the impact of the likelihood regulators on the waveform reconstruction, to understand the dependence of the waveform reconstruction on the value chosen for the Γ regulator. We start introducing the regulators from a theoretical point of view, to understand their role in the analysis and how they are tied to the waveform reconstruction. After this, show how these regulators can affect the sky-localization of the sources, and conclude with the discussion of thee values for Γ, specifically un-constrained likelihood ($\Gamma = 0$), a "soft-constrained" likelihood ($\Gamma = 0.6$) and a "hard-constrained" likelihood ($\Gamma = 1$).

6.3.1 Likelihood Regulators

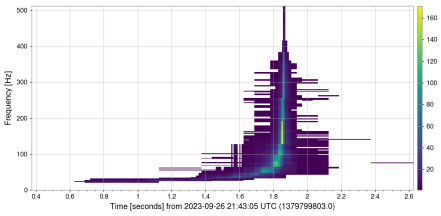
The regulators are two parameters (Γ and Δ) that were put in CoherentWaveBurst (cWB) to get rid of false alarms caused by ground noise. In here we focus on the Γ regulator, which is the one that has a direct impact on the waveform reconstruction. Some detector networks, like HL, are mostly sensitive to just one main gravitational-wave polarization (f_1). This means ground noise can easily create fake signals in the weaker cross-polarization (f_2) component. To fix this, we look at how well the detectors are aligned using the alignment factor α:

$$\alpha = \frac{|\mathbf{f}'_\times|^2}{|\mathbf{f}'_+|^2}, \quad 0 \leq \alpha \leq 1 \tag{6.3}$$

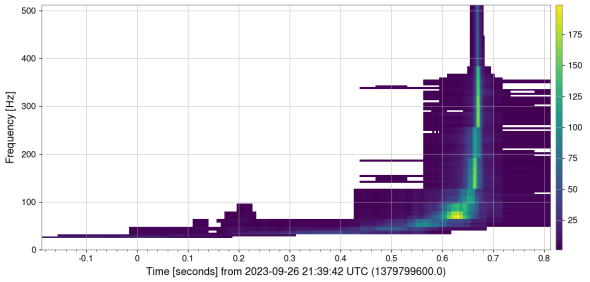
For the HL, α is usually very close to 0, which means the main polarization carries almost all of the signal energy. The Γ regulator uses this geometry to clear out the noise pixel by



(a) Likelihood time–frequency map using resolution levels between 4 and 10.



(b) Likelihood time–frequency map using resolution levels between 3 and 9.



(c) Likelihood time–frequency map using resolution levels between 2 and 8.

Figure 6.4: Likelihood time–frequency maps for the three resolution level configurations for one of the reconstructed events as a function of the resolution levels. Reducing the maximum resolution level produces more compact pixels in time, reducing their temporal spread and the corresponding leakage in the post-merger window.

pixel. Firstly, it calculates a polarization statistic called R:

$$R = \sqrt{e^2 + \sin^2 \gamma} \tag{6.4}$$

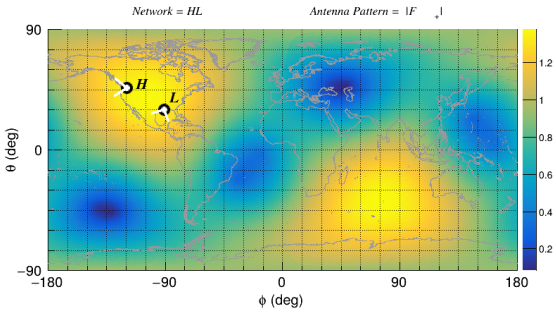
Here, e is the wave ellipticity and γ is the angle of the dominant polarization frame defined in equation 3.27. If a pixel is too noisy, it triggers this inequality:

$$R - 0.9 + \left(0.5 - \frac{1}{I_{\text{net}}}\right)(1-\alpha) - \frac{F_{\text{net}} \ln \Gamma}{(2I_{\text{evt}})^2} < 0, \tag{6.5}$$

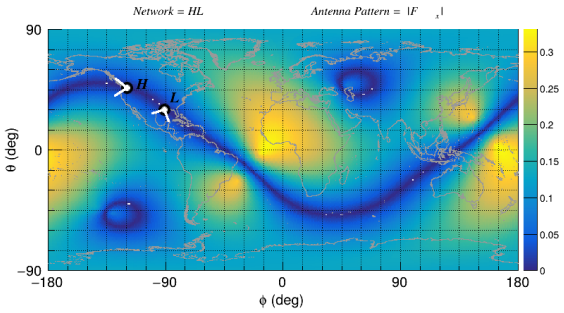
with I_{net} the effective number of detectors, I_{evt} the event index based on energy, and F_{net} the network sensitivity ($F_{\text{net}} = \sqrt{(f_1^2 + f_2^2)/(2I_{\text{net}})}$). At the same time, cWB checks a second condition to remove signals from parts of the sky where the network cannot see well:

$$|f_\times|^2 < \frac{\Gamma}{I_{\text{net}}}\left(I_{\text{evt}} - |f_\times|^2\right) \tag{6.6}$$

If a pixel meets these conditions, its cross-polarization components are set to zero to kill the noise. The values of Γ can range in the closed interval between 0 and 1. With $\Gamma = 1$ the pipeline cuts out noise very strictly, while with lower values Γ more information about the second polarization is retrieved, although more background noise will be included.



(a) The antenna pattern of the HL network for the first polarization f_1 of the dominant polarization frame.



(b) The antenna pattern of the HL network for the second polarization f_2 of the dominant polarization frame.

Figure 6.5: The antenna patterns of the HL network for the two polarizations of the dominant polarization frame.

6.3.2 The impact of the Likelihood Regulators

In coherent WaveBurst, the core part of the analysis is the computation of the likelihood, which is the central quantity used for both the detection and the reconstruction of the GW signals. In thix context, to improve the detection efficiency, the likelihood underwent several tuning and modifications, leading to the introduction of the so called "Likelihood Regulators" [72].
The likelihood regulators have been developed in the context of HL analysis to improve the detection efficiency of the pipeline, and are based on practical considerations. In particular, we focus on the Γ regulator, which is tied to the reconstruction of the f_2 polarization in the Dominant polarization frame (sec: 3.2.2), and can assume values in the range $\Gamma = [0, 1]$. We consider three different values of Γ: $\Gamma = 0$, the un-constrained, $\Gamma = 0.6$, a "soft-constrained" likelihood and $\Gamma = 1$, called "hard constrained" likelihood.

$\mathbf{\Gamma = 0}$ is the un-constrained Likelihood as it is derived from the purely statistical arguments, in which no regulator is applied to the likelihood. In partiuclar, $\gamma = 0$ allows for a full reconstruction of both the polarizations of the wave in the dominant polarization frame.

$\mathbf{\Gamma = 1}$ is the hard-constrained likelihood, in which the regulator has the maximum value and it is the standard value used for the LVK analysis. It has been shown that the hard constrained likelihood maximized the detection efficiency of the pipeline, providing a better rejection of noise transients. A second impact of the hard constrained likelihood is the complete rejection of the second polarization of the wave in the dominant polarization frame (see section 3.2.2). Rejecting a physical polarization may seem counter-intuitive at first, if not completely wrong, since it contains physical information about the source. However, being the two Ligo interferometers aligned, the network is mostly sensitive to the first polarization of the wave and almost completely blind to the second, subdominant polarization. The sensitivity of the HL network for the two polarizations in the dominant polarization frame is Figures 6.5a and 6.5b (notice the different scales wih $f_1 \in [0, 1.2]$ and $f_2 \in [0, 0.3]$). Because of the extremely loew sensitivity to the second polarization of the wave, the attempt to reconstruct it may end up with adding unnecessary noise to the reconstruction, which may be counter-productive and worsen our ability to reconstruct the waveform accurately.

$\Gamma = 0.6$ is a third value for the regulator, which we call "soft-constrained" likelihood. We chose a third value to see how the waveform is impacted with the usage of a different value for this parameter. Among all the possible values, $\Gamma = 0.6$ has been found to minimize the error in the sky localization for the LHV analysis [81]. For this reason, even if the setting is different, we report the results on the reconstruction for this specific value.

For the following analysis, the set of injection is the same one used for the all-sky comparison of CBC signals on the LH network. Only the commonly reconstructed events between analysis are considered.

Sky localization

The sky localization is a huge topic in itself and is not the main focus of this thesis. But since it is related to the likelihood regulators, it is interesting to see the two are related. Also, being both the waveform and the sky position estimated by maximizing the same likelihood, there may be some correlations between the accuracy in the sky-localization and the accuracy in the waveform reconstruction. For the HL network, the sky-localization is performed by maximizing the Likelihood over the sky positions. The choice of two detector networks is not ideal for the sky localization, since it is mostly drive by the time delay between the two detectors, and is therefore only able to constrain the source to an annulus in the sky. The optimal choice would be to perform the analysis on the HLV network, the interested reader can find a whole-discussion of HLV network in [81].

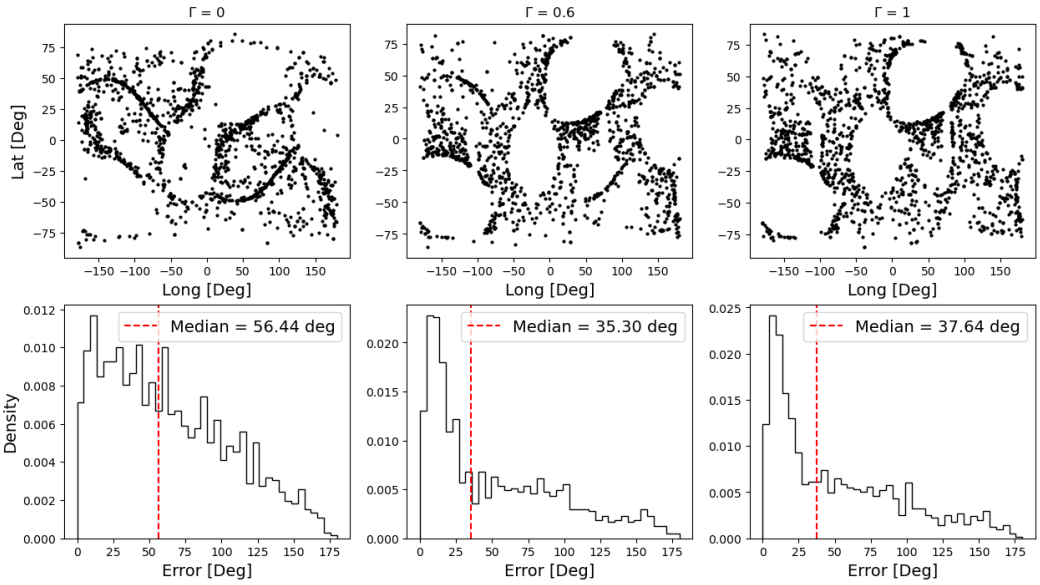


Figure 6.6: Sky localization of the sources for the three values of Γ. The results are displayed, from left to right, for $\Gamma = 0$ (un-constrained likelihood), $\Gamma = 0.6$ (soft-constrained likelihood) and $\Gamma = 1$ (hard-constrained likelihood). The top plot, shows the reconstructed positions for the injected sources. The bottom plot shows the histogram of the errors between the injected sky location and the reconstructed sky location, computed as the angular distance between the two points in the sky.

For the two detectors networks, the results for the sky localizations are reported in Figure 6.6. They are reported as an histogram of the errors between the injected sky location and the reconstructed sky location, computed as the angular distance between the two points in the sky. Since two angles are defined between the two directions, the error is defined as the minimum between the two angles, so that the maximum error is 180 degrees. From left to right, the plot shows the results for $\Gamma = 0$ (un-constrained likelihood), $\Gamma = 0.6$ (soft-constrained likelihood) and $\Gamma = 1$ (hard-constrained likelihood) respectively. The top plot shows the reconstructed positioins for the injected sources. We can notice some relevant differences between the three plots. None of them actually recontructs any event inside the blind spots of the HL network, but the contour tend to change as Γ increases. For the $\Gamma = 0$ the contours are the faintest, while they become more defined for the soft constrained and hard constrained likelihood. There is no visual difference on the lobes for the two reconstruction. There is another evident difference among the three plots. For $\Gamma = 0$, there is a non negligible number of events that are reconstructed along the line joining the two detectors, i.e. the line along which the network is completely blind to the second polarization of the wave. Increasing values of Γ progressively reject the second polarization, and the number of events reconstructed along that line drops, until any relative excess dissaperas for $\Gamma = 1$.
The Bottom line of the figure shows the histogram of the errors between the injected sky location and the reconstructed sky location, computed as the angular distance between the two points in the sky. The results for the sky-localization show that the choice of the regulator has a significant impact on the sky-localization of the sources, with $\Gamma = 0$ providing the worst sky-localization, with a median error of ~ 56.4 degrees, and a profile for the error distribution that drops slowly and almost linearly up to 180 degrees. The two values $\Gamma = 0.6$ and $\Gamma = 1$ provide a significant improvement in the sky-localization, with a median error of ~ 35.3 degrees for $\Gamma = 0.6$ and ~ 37.6 degrees for $\Gamma = 1$. The two distributions look extremely similar, with a slightly larger bulk of event reconstructed with an error smaller than ~ 25 degrees. The second part of the distribution dropping slowly up to 180 degrees without any significant differences between the two distributions.
Interestingly, forcing the likelihood to neglect a second polarization in a network which is completely blind to it, provides a significant improvement in the sky-localization of the sources. The attempt to reconstruct the second, undetectable, polarization, instead, provides a significant degradation of the sky-localization capabilities of the pipeline.

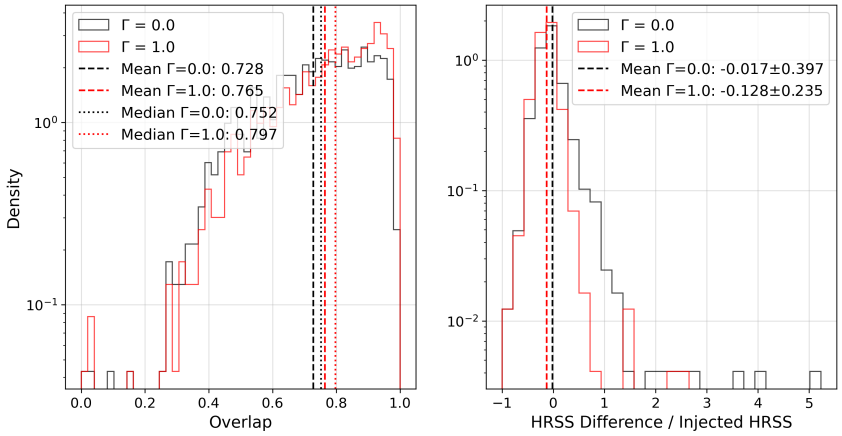
Waveform reconstruction

We now switch to a second perspective for the likelihood regulators, analyzing their impact on the waveform reconstruction. The waveform reconstruction is performed by maximizing the same likelihood used for the sky-localization, so it is expected that the choice of the regulator to impact the waveform reconstruction as well. In particular, we make pairwise comparison of the reconstruction for the various regulators, focusing on the two comparison $\Gamma = 0$ VS $\Gamma = 1$ and $\Gamma = 0.6$ VS $\Gamma = 1$. This analysis exploits the exact same set of injections used for the sky-localization, so that the results can also be put together in a second moment. As a metric to assess the quality of the waveform reconstruction, we use the overlap between the reconstructed waveform and the injected waveform. To compute the overlap between the two waveforms, the reconstructed waveform is shifted in time to be aligned with the injected waveform. Together with the overlap (intrinsically insensitive to the relative amplitude of the two signals), we also look at the error in the hrss of the

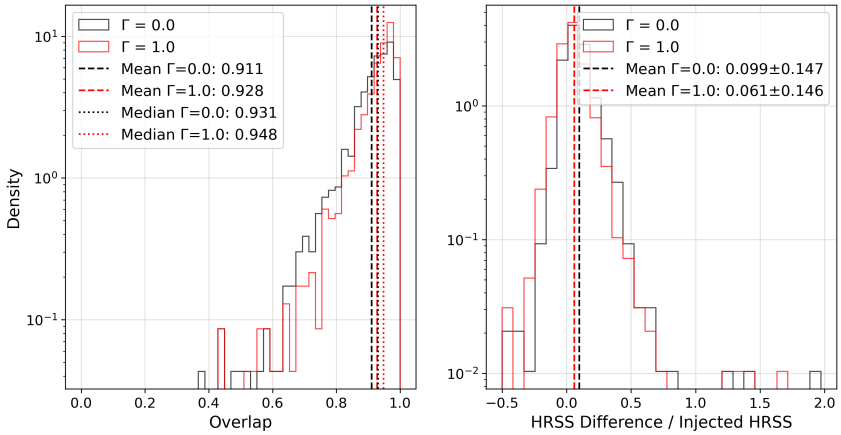
reconstructed waveform:

$$\frac{hrss_{reconstructed} - hrss_{injected}}{hrss_{injected}} \tag{6.7}$$

or the same for the SNR in the whitened domain. For the analysis, only the common triggers between the two method are considered, so that the comparison can be performed event by event. Figures 6.7a and 6.7b show the comparison between the two methods for $\Gamma = 0$ and



(a) Overlap and hrss comparison for strain data.



(b) Overlap and hrss comparison for whitened data.

Figure 6.7: Comparison of the overlap and hrss error between the reconstructed waveform and the injected waveform for $\Gamma = 0$ and $\Gamma = 1$. The upper plot shows the results for the strain data, while the lower plot shows the results for the whitened data.

$\Gamma = 1$ for the strain data and the whitened data respectively. Im the strain domain (upper plot), the distribution of the overlap (left plot) for $\Gamma = 1$ is shifted toward larger values of the overlap, with an increase in both the mean and the median value of ~ 4 percentage points. The difference in total hrss, on the right, shows that both the distributions for $\Gamma = 0$ and $\Gamma = 1$ have got a mean that compatible with zero whithin one standard deviation for both methods. On the left side of the mean, the two distributions are almost identical. On

the right side, instead, the distribution for $\Gamma = 0$ is more skewed toward larger values of the error, with a larger number of events in which the reconstructed hrss is larger than the injected one. This is also reflected in the larger standard deviation for $\Gamma = 0$ compared to $\Gamma = 1$. In the whitened domain (lower plot), the distribution of the overlap for $\Gamma = 1$ is still shifted toward larger values of the overlap, with an increase in both the mean and the median value of ~ 1.7 percentage points. The distribution of the hrss error, instead, is now more similar for the two cases. Again, both mean are compatible with zero within one standard deviation, and the point estimate for the mean of $\Gamma = 1$ is slightly smaller that the one for $\Gamma = 0$. The shape of the two distribution is now more similar in the two cases, with the distribution for $\Gamma = 1$ slightly shfted towards the right, with a larger number of events in which the reconstructed hrss is larger than the injected one.

In both the whitened and the strain domain, the hard-constrained likelihood ($\Gamma = 1$) provides a better reconstruction of the waveform, with a larger value for the overlap and a smaller error in the hrss of the reconstructed waveform. Again, as for the sky localization, this behavior can be explained by the fact that, using no regulator, the pipeline tries to reconstruct a second polarization that is not detectable by the network. In this attempt, the pipeline forces noise to be reconstructed as a second polarization, which can lead to an overestimation of the hrss of the reconstructed waveform, and therefore to a larger number of instances in which the reconstructed hrss is larger than the injected one. In the whitened domain this effect is mitigated by the fact that the noise fluctuations are suppressed at smaller SNRs, leading to an overall improvement in both the overlap and the hrss error for both methods.

To show this in deeper detail, we select two events, one from the strain domain and one from the whitened domain. The selected events are those for which the difference in the overlap between the two methods is maximized. We show the reconstructed waveforms for the two methods in Figure 6.8. The values for the overlap of the reconstructed waveforms are reported in Table 6.2.

	ovlp	
	Gamma = 0	Gamma = 1
Strain	0.3585	0.9358
Realigned	0.5451	0.9358
Whitened	0.6631	0.9038

Table 6.2: Overlap values for different configurations.

The top plot shows reconstructed waveforms for $\Gamma = 0$ and $\Gamma = 1$ in the strain domain. The overlap values differ dramatically: overlap($\Gamma = 1$) $\simeq 0.94$ versus overlap($\Gamma = 0$) $\simeq 0.36$. The small value for $\Gamma = 0$ is firstly due to a wrong alignment of the two waveforms. The reconstructed waveform for $\Gamma = 0$ exhibit such a large amount of noise in the inspiral phase that the time correlation is maximized by the noise in the inspiral phase rather than by the merger-ringdown phases. Notably, the early inspiral phase for $\Gamma = 0$ exhibits fluctuations with energy exceeding that of the ringdown phase.

When the $\Gamma = 0$ waveform is realigned to be synchronized with the merger (center plot), the overlap improves to ovlp($\Gamma = 0$) $\simeq 0.545$. This demonstrates that noise inclusion creates an alignment artifact affecting the overlap, but even accounting for this effect, noise inclu-

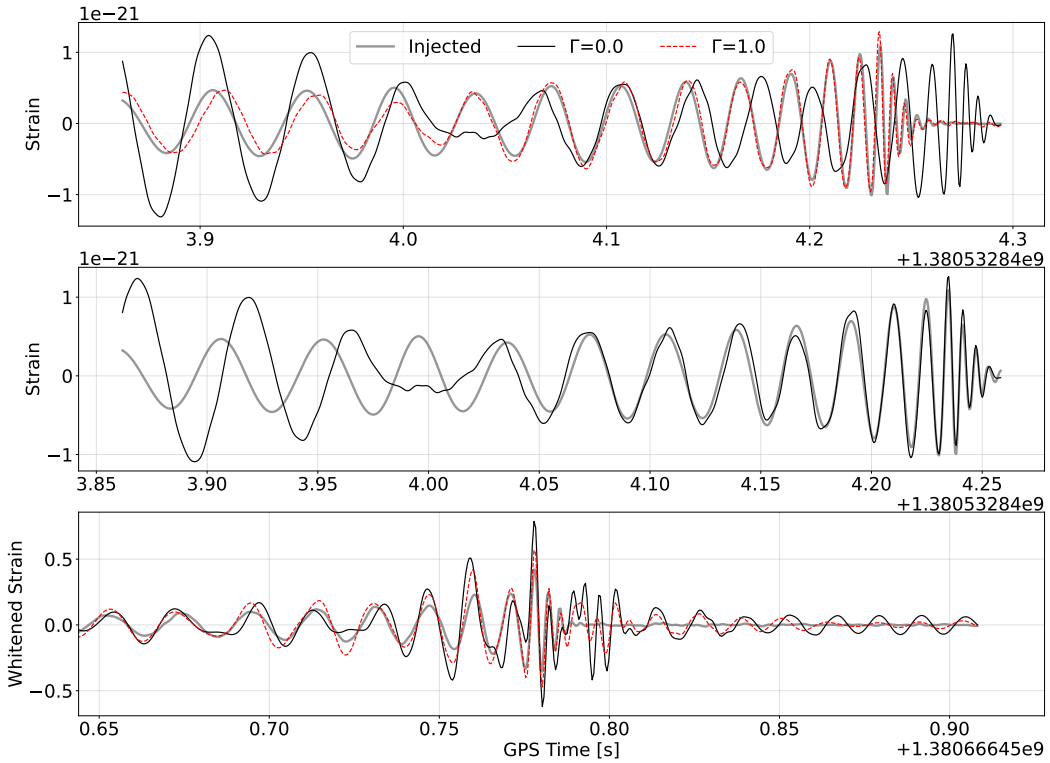


Figure 6.8: Comparison of the reconstructed waveforms for $\Gamma = 0$ and $\Gamma = 1$ for a specific event that maximizes the difference in the overlap between the two methods. The upper plot shows the waveforms in the strain domain. Due to large amount of noise for $\Gamma = 0$, the cross-correlation is maximized at the inspiral. The central plot correct for this estimate. The bottom plot shows the waveforms in the whitened domain.

sion degrades overall reconstruction quality.
The bottom plot shows the same comparison in the whitened domain. Here, the merger-alignment phenomenon vanishes, yet $\Gamma = 0$ still yields an overlap approximately 24 percentage points lower than the hard-constrained case. While the reconstruction envelopes remain similar, the unconstrained reconstruction contains fluctuations that increase mismatch with the injected waveform, resulting in poorer overall signal reconstruction.

$\Gamma = 0.6$ **and** $\Gamma = 1$

For the two values $\Gamma = 0.6$ and $\Gamma = 1$, we propose the exact same comparison we performed on the previous case, with the distribution of the overlaps and hrss error for the two methods. The results are reported in Figure 6.9. From left to right, the results for the overlap and *hrss* error are reported, firstly for the strain data and then for the whitened data. The results show that the two methods provide an almost identical distribution for the two tuning of the parameter, with no visible difference between the two disttributions. The dashed, vertical line in each plot indicates the mean for the distribution for both the $\Gamma = 0.6$ and $\Gamma = 1$ cases, in both the mean of the reconstructed waveforms are identical, and a binning error which is fully compatible within the poisson fluctiations of each individual bin. We can then

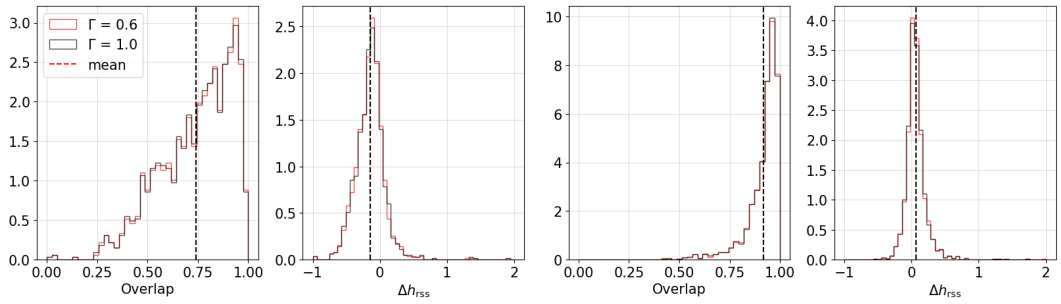


Figure 6.9: Overall caption describing both panels.

fastly conclude that, in this specific case, there is no reason to prefere one value of Γ over the other, even though these results may be limite by the fact that we are only selecting the events that are common between the two methods. Since we know that for $\Gamma = 1$ the detection efficiency for the HL network is maximized, while there is no significant difference in both the sky localization and the waveform reconstruction, we can conclude that $\Gamma = 1$ is the optimal choice for the analysis, and it will be used as the default value for the rest of the thesis.

6.3.3 Correlation with the sky-localization

In the previous discussion, the only motivation we gave for the different value of the overlap in the different cases was the inclusion of noise in the reconstruction for $\Gamma = 0$. But, as we said in the introduction, both the sky-localization and the waveform reconstruction are performed by the likelihood maximisation, so that one could expect some level of correlation between the two variables. It is credible that the slighlty lost overlap displayed by the $\Gamma = 0$ can be related to the worse sky-localization capabilitied of the method. To investigate this correlation, we provide a box plot of the overlap as a function of the error in the sky localization with bins of 20 degrees.

The results are shown in Figure 6.10 for the strain domain (but the whitened domain gives similar results). We firstly recognise the systematically larger value of the overlap for $\Gamma = 1$ with respect to $\Gamma = 0$ for all the bins, confirming the results of the previous analysis. The last bin is the only exception, but it is severly under-populated that its results are not statistically significant. None of the plots exhibit a clear correlation between the overlap and the sky error, meaning that our reconstruction is "robust" with respect to the sky localization. In other words, the total likelihood can be imagined as factorised into the two "sky-localization" and "waveform reconstruction" likelihoods.
This is actually a good and reasonable result for a burst pipeline: if our reconstruction was strongly correlated with the sky-localization, it would mean that there should be some level of information about the morphology of the signal leaking from the sky location to the reconstruction. But since we aim to be completely agnostic with respect to the morphology of the signal, we can expect any possible morphology coming from every possible sky direction.

This last plot can make us conclude that most of the difference in the two configurations are actually arising from the second polarisation component of the wave rather than from a correlation with the error in the sky-localization.

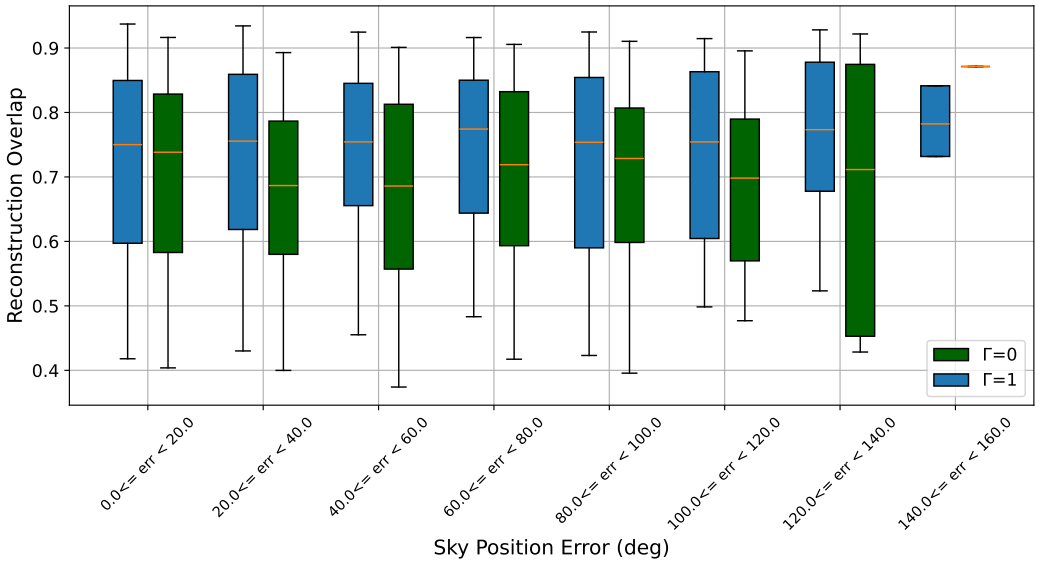


Figure 6.10: Box plot of the overlap as a function of the error in the sky localization for $\Gamma = 0$ and $\Gamma = 1$. The x-axis is binned in intervals of 20 degrees, while the y-axis represents the distribution of the overlap for each bin. The box plot shows the median (horizontal line), the interquartile range (box), and the whiskers representing the range of the data.

6.4 Conclusions

In this chapter we compared the standard cWB whitening method and the MESA whitening method, focusing on two main aspects: the accuracy of the reconstructed waveform and the SNR leakage in the post merger window. A further element considered in the comparison is the value of the WDM parameter β that controls the shape of the wavelet.
For this reason, we performed a comprehensive analysis of the reconstructed waveforms, by injecting both CBC signals and a variety of burst signals into the data and comparing the reconstructed waveforms with the injected ones.
The advantages seen for MESA in chapter 5 at the whitening stage are largely mitigated once the reconstruction stage is considered.
For leakage, the most impactful parameters are those associated with the WDM transform used in pycWB's multi-resolution analysis—namely the β parameter and the resolution levels (especially the maximum level l_{high}). This is evident in Figure 6.3, where the leakage behavior changes significantly when the maximum resolution level is modified; β also plays an important role, with $\beta = 2$ producing systematically smaller leakage than $\beta = 6$, especially when cWB whitening is used.
This does not mean that whitening plays no role in the leakage of the reconstructed waveform; rather, its effect is sub-dominant with respect to the contamination arising from neighboring pixels in the time–frequency representation.
For the fidelity of the reconstructed waveform, we find a similar scenario: most improvements come from reducing β from 6 to 2, while the results do not show a dependence on the whitening method, with MESA and pycWB performing similarly. Overall, we find strong evidence that most improvements in reconstruction arise from adopting a WDM transform

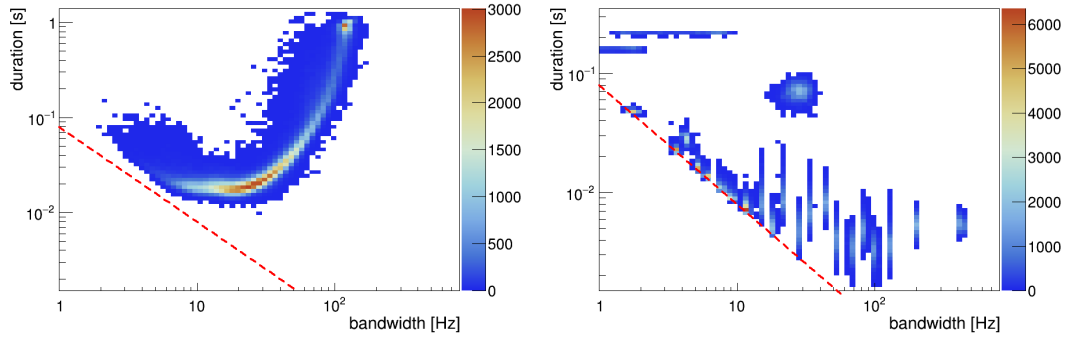


Figure 6.11: Distribution of reconstructed duration and bandwidth for the injected signals, zero–noise and whitened. Left panel shows the time-frequency representation for the CBC injections, while the right panel shows the same for the burst injections. The red-dashed line represents the Gabor–Heisenberg limit. This figure is taken from [87].

closer to the Gabor limit, enabling a more compact time–frequency representation of the signal.

One could argue that WDM, being a reversible transform, should not distort the signal representation since all information is preserved in the pixel basis. In practice this is not true, because the multi-resolution analysis performed in cWB is not a perfect representation: only pixels with the highest energy content are selected, while the rest are discarded. It is therefore crucial that selected pixels are as compact as possible in both time and frequency, minimizing resolution errors and ensuring that discarded pixels do not contain relevant signal information. For burst signals, the distribution of the overlap between the reconstructed and the injected waveform improves when moving the WDM transform closer to the Heisenberg-Gabor limit, while no relevant improvement has been found in the CBC analysis. This can be understood by considering the durations and frequency bandwidths of CBC and burst signals, reported in Figure 6.11. For CBC signals, since they are typically not close to the Gabor–Heisenberg limit (represented by the red dashed line), a WDM with $\beta = 6$ is sufficient to capture their prominent features. This is no longer true for the several burst signals which lie close to the Gabor–Heisenberg limit; here, a WDM with $\beta = 2$ is a more suitable choice to reconstruct the relevant features.

In a general, we didn't study the impact of β on the detection efficiency. While we do not expect significant differences as a function of the β parameter, in case they are present one could choose the β value which maximizes the detection efficiency and follow-up the significant candidates with $\beta = 2$, whenever a better time resolution of the spectral analysis is beneficial.

After studying the impact of the whitening and WDM on the waveform reconstruction, we switched to the study of the likelihood regulators, which are a set of parameters introduced in cWB to improve the detection efficiency of the pipeline. In particular, we studied three different values of the Γ regulator, which is tied to the reconstruction of the second polarization of the wave in the dominant polarization frame. We showed that the choice of the regulator has an impact on the quality of the reconstruction, with the default value $\Gamma = 1$ (hard-constrained likelihood) providing on average the best reconstruction of the waveform for the LH analysis.

Chapter 7

Bootstrap Uncertainty in Burst Waveform Reconstruction

7.1 Introduction

In the previous chapter, we have delved into the details of the waveform reconstruction process into pycWB. We have mostly focused our attention on the impact of the whitening process and the tuning of the wavelet parameters on the final, reconstructed waveform. In the previous chapter, all the performed tests were applied on the "point estimate" of the waveform over time, without any confidence intervals. In this section, we want to find a robust method to build confidence intervals around the pycWB estimate of the waveform with no information on the signal.
To do so, we will rely on the bootstrap method, firstly introduced by Bradley Efron in 1979, which is statistical technique used to estimate the sampling distribution of a statistic by resampling with replacement from the original data. There are two formulations of the bootstrap method: the parametric bootstrap and the non-parametric bootstrap. The non-parametric bootstrap is more commonly used and does not require any assumptions about the underlying population distribution. The parametric bootstrap, instead, relies on the assumption that the data is drawn from a specific parametric family of distributions, and it uses the estimated parameters of this distribution to generate new bootstrap samples. Both methods are powerful tools for estimating the variability of the statistics of interest. In this chapter the bootstrap method for building confidence interval is introduced. We will first introduce the method in a simple case scenario, where we want to estimate the error on some simple statistic for some data. Then, we will try to transfer the bootstrap method to the more complex case of time series. Finally, we will apply the method to the case of burst waveform reconstruction in pycWB.

7.2 Bootstrap method

The bootstrap method start from the idea that we can resample the original, observed dataset to make inference on a statistic. The bootstrap method is particularly useful when dealing with small sample sizes or when the theoretical distribution of the statistic is un-

known. Suppose we have a dataset of n observations, i.e.

$$F \to d = \{d_1, d_2, \ldots, d_n\} \tag{7.1}$$

in which d_i are independent and identically distributed (i.i.d.) random variables drawn from an unknown distribution F. A complete description for the data is achieved if we know the real distribution F of the data. Once the distribution for the data is known, the quantities of interest are usually some function of the distribution F. Such quantities are called parameters, and encode some partial (or, in some cases, complete) information about the distribution F. Given the distribution F, a general parameter of interest can be defined as a function of the distribution F as follows:

$$\theta = t(F) \tag{7.2}$$

In general, one does not know the real distribution F, but can get some information about F using the observed data building the empirical distribution $\hat{F}$. The empirical distribution $\hat{F}$ is a discrete distribution that assigns equal probability to each observation in the dataset. Therefore, it associates a probability to a subspace A of the real distribution F as follows:

$$\hat{F}(A) = \frac{1}{n} \sum_{i=1}^{n} I(x_i \in A) \tag{7.3}$$

where I is the indicator function which equals 1 if $x_i \in A$ and 0 otherwise, which can be used as a proxy for the real distribution F.
In a real case scenario, to compute the parameter of interest θ, one estimates it directly from the empirical distribution $\hat{F}$ rather than from the distribution F. This is done by replacing F with $\hat{F}$ in the definition of the parameter θ, which gives us an estimate of the parameter, denoted as $\hat{\theta}$:

$$\hat{\theta} = t(\hat{F}) = F(d). \tag{7.4}$$

The equation above is usually referred to as the **plug-in principle**, and it is the basis of the bootstrap method. The estimate for the parameters is then computed as a function of the observed, available data. The theory on how to build these estimator is a large field of statistics, and it is not the focus of this chapter.
The estimate of the parameter $\hat{\theta}$ itself is a random variable, since it depends on the observed data, which is itself a random variable, so that to retrieve information about the observed data, a point estimate for the parameter of interest is barely enough. What we really need to draw conclusion from this estimate, is to have some information about the variability, or the error, of the estimate $\hat{\theta}$. In some cases, such errors can be drawn with ease by the construction of the estimator itself. One example is the error on the mean of normally distributed samples. On the contrary, if the statistic of interest is more complex, finding a reasonable estimate for the error of the estimator can be a very hard task. The bootstrap method is a powerful tool to estimate the variability of an estimator, which takes advantage of the construction of the empirical distribution $\hat{F}$ as a proxy for the real distribution F, together with the plug-in principle.
The bootstrap method relies on the idea of resampling from the empirical distribution $\hat{F}$ to create new datasets, called bootstrap samples. Each bootstrap sample is generated by randomly sampling n times (with $n = len(d)$) **with replacement** from the original dataset d, obtaining a bootstrap replicate for the data

$$F^* \to d^* = \{d_1^*, d_2^*, \ldots, d_n^*\}. \tag{7.5}$$

Notice that in the bootstrap sample, some of the original observations may be repeated, while others may be left out. We can notice that, building a bootstrap replicate is equivalent to sampling from the empirical distribution $\hat{F}$ rather than the original, unknown distribution F. Each bootstrap sample is then used to compute a bootstrap replicate of the parameter of interest, denoted as $\hat{\theta}^* = f(d^*)$, with $f(.)$ being the same function used to compute the original estimate $\hat{\theta}$. This process is repeated a large number of times B, with B known as the number of bootstrap samples, and it results in a distribution of bootstrap replicates for the parameter of interest

$$\hat{\theta}^* = \{\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^*\}. \tag{7.6}$$

This procedure is known as "non-parametric bootstrap", since it does not make any assumption about the underlying distribution of the data, and it relies solely on the empirical distribution $\hat{F}$ as a proxy for the real distribution F. We now want to introduce another version of the bootstrap method, which is defined as the "parametric bootstrap".

The parametric bootstrap

The parametric bootstrap relies on the assumption that the data is drawn from a specific parametric family of distributions, and it uses the estimated parameters of this distribution to generate new bootstrap samples. Instead of using the pure, empirical distribution $\hat{F}$ as a proxy, the parametric bootstrap assumes a parametric model for the distribution F_{par}, whose parameter values are estimated from the data. What differs from the previous example, is that the bootstrap samples are now generated by sampling from the parametric distribution F_{par}, rather than the empirical distribution $\hat{F}$.

$$F_{par} \rightarrow d^* = \{d_1^*, d_2^*, \ldots, d_n^*\}. \tag{7.7}$$

Again, from this sampling, we can compute the bootstrap replicates for the parameters of interest $\hat{\theta}_{par}^*$ and we can use the distribution of the bootstrap replicates to estimate the variability of the original estimate $\hat{\theta}$. Both the bootstrap methods have their own advantages ad disadvantages. The non-parametric bootstrap is more flexible, since it does not make any assumption about the underlying distribution of the data. The parametric bootstrap, on the other hand, can be more efficient if the assumed parametric model is correct, but it can lead to biased estimates if the model is mis-specified. The parametric bootstrap usually assumes a normal distribution for the data. In this case, it can give a good approximation of standard, statistical formulas usually obtained under the assumption of normality.

Once the distribution of the bootstrap replicates is obtained with either method, we can use it to estimate the variability of the original estimate $\hat{\theta}$. In the thesis we will mainly use two methods, the computation of the percentiles over the ensemble of the reconstructed waveforms and the standard deviation of the bootstrap replicates. There are better ways to estimate the error from the data, but they are ill-defined in the context of time-series bootstrap, so that we won't investigate them any further. [1].

[1] A more advanced alternative is the Bias Corrected and Accelerated (BCa) method, which accounts for both bias and skewness in the bootstrap distribution by adjusting its quantiles. It is more robust than the standard percentile method, but it is also more computationally expensive. Unfortunately, it is not applicable in the case of time series data, because it relies on the Jackknife estimate, which is not defined in this context.

7.2.1 Bootstrap regression

All the example obtained so far have been based on the observation of i.i.d. data, which are an extremely important, but only cover a reduced number of cases.
In many cases of interest, we are not dealing with simple i.i.d. data, but with more complex data in which two variables are related on another via some functional relationship, let's say

$$d = f(x, \alpha) \tag{7.8}$$

for some function f and some parameters α. In this case, we do not define the data as simply a vector of observations d, but as a set of pairs of observations:

$$D = \{(x_1, d_1), (x_2, d_2), \ldots, (x_n, d_n)\}. \tag{7.9}$$

In this case, we are interested in estimating the parameters α of the function f that best fits the data.. Usually, one does not observe the data directly, but rather a noisy version of it, which can be expressed as

$$d_i = f(x_i, \alpha) + \epsilon_i \tag{7.10}$$

with ϵ_i being the noise term, which is usually assumed to be i.i.d. and normally distributed with zero mean and some variance σ^2. In our cases, we consider f to be a deterministic function of the variable x and the parameters α. Under this conditions, if the functional form of f was known, the problem would be solved with parametric approaches, such as Bayesian Parameter Estimation and Monte Carlo techniques, as it is done for CBC templates searches (See Section 2.3.2). This is no more true for the case of burst waveform, in which no assumption on the functional form of f is made and it is estimated directly from the data using equation 3.31 and 3.32. For the regression problem, the bootstrap method can be mainly applied in two different ways: the residuals bootstrap and the pairs bootstrap [52]. We will only focus on the first one, since it is easily convertible to time-series.
Given a regression problem, what one can do is fitting the model to the data, obtaining an estimate for the parameters $\hat{\alpha}$, and then computing the residuals of the fit as follows:

$$\hat{\epsilon}_i = d_i - f(x_i, \hat{\alpha}) \tag{7.11}$$

The residuals $\hat{\epsilon}_i$ are not the "real" stochastic noise underlying the estimate, but rather an estimate for the error ϵ_i in the data, and that they are homoscedastic and independently distributed. Their representativeness of the real error ϵ depends on the quality of the fit and the model specification. If the residuals are homoscedastic and independently distributed, then they can be used as a proxy for the real error ϵ_i, and we can apply the bootstrap method to the residuals, as we did for the data at the beginning of the chapter.
In this case, given a "meaningful" estimator for the α, the distribution of central importance is not the distribution of the data, but rather the distribution of the error F_e, which is unknown. In this context then, the bootstrap can be applied in both its parametric and non-parametric version in the same way as before, by using the residuals $\hat{\epsilon}_i$ as a proxy for the real error ϵ_i. We can think of the series of the error as being drawn from some unknown distribution F_ϵ, whose empirical distribution $\hat{F}$ can be used to build the bootstrap samples for the error, which are denoted as ϵ^*. The bootstrap replicates for the error are then defined as:

$$\hat{F}_\epsilon \rightarrow \hat{\epsilon}^* = \{\hat{\epsilon}_1^*, \hat{\epsilon}_2^*, \ldots, \hat{\epsilon}_n^*\} \tag{7.12}$$

The bootstrap replicates for the data are then defined as:

$$d^* = f(x, \hat{\alpha}) + \epsilon^* \tag{7.13}$$

with $\hat{\alpha}$ being the "plug-in" estimate for the parameters α, and the data d^* can be used to obtain a series of bootstrap replicates for the parameters $\hat{\alpha}^*$, and can then be used to estimate the variability of the original estimate $\hat{\alpha}$. With the regression bootstrap, we shifted our focus from the distribution of the data to the distribution of the errors.
Notice that this results offers an interesting parallelism with what happens in the standard Bayesian methods for parameter estimation. In here, we are using bootstrap to make inference on $\hat{F}_e$ and use it as a proxy of the original distribution, but we are not in any way making any explicit reference to the distribution of the data. But, what happens is the Bayesian context of parameter estimation, where the likelihood is defined, one can show that the Likelihood is fully determined by the distribution of noise. Moreover, the Whittle Likelihood is the likelihood for the noise **computed on the residuals** between the data and the model to be fit. Also in this case, what matters is the distribution of the error, and the data can be seen as a "random" realization of the error, since the model is fixed and deterministic. What we are doing here is extremely similar, we firstly find a "meaningful" estimate for the parameters $\hat{\alpha}$, and then we add several realizations of the error, still intended as the value of the residuals between the data and the best fit. .

7.3 Time-Series bootstrap

We now introduce the bootstrap in the timeseries context. As we have seen before, the bootstrap can be used to find the variability of an estimator for the parameters of a regression problem. In these cases, we are assuming the errors to be i.i.d. In many cases of interest, the errors are not i.i.d, but they are correlated. This is the case for time series, in which the error at a given time is usually correlated with the error at another time. In this case, the standard bootstrap method cannot be applied, since it relies on the assumption of i.i.d. data. However, there are several variations of the bootstrap method that have been developed to handle time series data, which are known as time-series bootstrap methods [58]. These methods have been developed provide bootstrap representations of the time series of the data

$$d(t) = d_{t_1}, d_{t_2}, \ldots, d_{t_n} \tag{7.14}$$

with t being the time index. Our goal is not to build a bootstrap representation for the time series, but finding the variability of an estimator over the time series. In this case, we are working in the data hypothesis:

$$d(t) = f(t, \alpha) + \epsilon(t) = h(t) + \epsilon(t) \tag{7.15}$$

and the parameters to be estimated are the value of the function h at every time t. Our problem in this case is to find the variability of the estimator $\hat{h}(t)$ for the signal $h(t)$, which is a function of time. Again, as we did for the regression problem, instead of sampling from the data, we can sample from the empirical distribution of the error and create new bootstrap replicates as:

$$d^*(t) = \hat{h}(t) + \epsilon^*(t) \tag{7.16}$$

with $\hat{h}(t)$ being the on-source estimate for the signal, defined by equation 3.31 and 3.32, and projected onto the detector antenna patterns. For time series, to sample from the error distribution AR-Sieve bootstrap (parametric), or the Block-Bootstrap (non parametric). The AR-Sieve bootstrap relies on the assumption that the error is generated by an autoregressive process, so that an $AR(p)$ process is fit to the residuals to reproduce its variability. The 'Block Bootstrap' instead is tuned in such a way that the correlation structure of the

residual is preserved when they are resampled. Both the methods are discussed in [58, 52].

Parametric AR-Sieve Bootstrap

The AR-Sieve bootstrap relies on the assumption that the time series under investigation can be modeled as a stationary process generated by an autoregressive process of infinite order $AR(\infty)$:

$$d(t) - \hat{h}(t) = \epsilon_t = \sum_{i=1}^{\infty} \phi_i \epsilon_{t-i} + e_t \tag{7.17}$$

where e_t represents the innovations from a white noise process. The AR-Sieve approximation is obtained by fitting a finite autoregressive process of order p to the data, where p is a hyperparameter that can be tuned using an information criterion, such as the Akaike Information Criterion (AIC) or the Final Prediction Error (FPE) [88]. In our case, the estimate of the autoregressive parameters is obtained via MESA (see Sec. 4.1). The fitted autoregressive process is then used to generate new bootstrap replicates for the error, which are subsequently used to reconstruct the bootstrapped data.

In its standard formulation, the AR-Sieve bootstrap computes the estimated residuals of the process ϵ_t as:

$$e_t = \epsilon_t - \sum_{i=1}^{p} \hat{\phi}_i \epsilon_{t-i}. \tag{7.18}$$

The bootstrap replicates are then generated by the fitted autoregressive process, where the random innovations are obtained by resampling the empirical residuals e_t with replacement. In practice, the classical AR-Sieve bootstrap whitens the stochastic process ϵ_t using the fitted autoregressive model and resamples the errors directly from these whitened residuals, providing realizations without making a priori assumptions about the underlying distribution of e_t.

For our purposes, since pycwb operates in the whitened domain, a non-parametric version of this bootstrap arises more naturally, which we introduce in the following section. Consequently, in this section we work with a parametric formulation of the AR-Sieve framework that relies on assuming a specific distribution for the innovations e_t. Given our use of MESA, we assume the innovations are normally distributed and utilize the estimated residual variance $\hat{\sigma}_e^2$ to sample new innovations from a Gaussian distribution.

Instead of directly applying the fitted autoregressive process in the time domain, we implement an equivalent, parameterized representation in terms of the power spectral density (PSD). Following the approach by Timmer and König [112], the error series is simulated by sampling directly from the frequency-domain distribution:

$$\epsilon^*(f) \sim \mathcal{N}(0, \hat{S}_\epsilon(f)) \tag{7.19}$$

and subsequently transforming back to the time domain. Under the assumption that $e(t)$ is normal and i.i.d., the time-domain autoregressive representation and this frequency-domain PSD representation are statistically equivalent.

By proceeding this way, we enforce the assumptions of stationarity and normality on the error, which may not hold globally for gravitational-wave noise. However, for short transients—where we are primarily interested in characterizing the statistical fluctuations of the noise at the exact time of the signal rather than its long-term evolution over time—these

assumptions are reasonable and widely adopted in the literature. While the assumption of normal innovations can be relaxed to accommodate other parametric distributions, we do not explore these alternatives further in this work. For a fully distribution-free approach, one can instead employ the non-parametric bootstrap detailed in the next section.

Block Bootstrap

The block bootstrap is a non-parametric method that relies on the idea of resampling from the residuals to preserve the time correlation structure of the error. The idea is to divide the residuals into blocks of length l, where l is large enough to capture the correlation of the error, and small enough to have a sufficient number of blocks to resample from. The blocks are then resampled with replacement to create new bootstrap replicates for the error, which are then used to build new bootstrap replicates for the data. In case of gravitational waves data, the block bootstrap is not the most suitable choice for the two following reason:

- The noise displays a long correlation structure, which would require very long blocks to be captured
- The noise is usually non-stationary, and the main assumption of the bootstrap for the data to be i.i.d is not satisfied.

For these two reasons, we implement the non parametric bootstrap to resample directly from the whitened residuals $e_W(t)$, which are expected to be independent by construction. In this domain the non-parametric bootstrap reduces to the simple regression case.

7.4 Bootstrap in pycWB - Examples

The above discussion introduces the bootstrap method and how it is used to build confidence intervals. In this section, we will introduce how the bootstrap is now implemented and used in pycWB to build confidence intervals.
To build the confidence intervals, two different analysis are required: the on-source analysis, which is the standard pycWB analysis on the original data, which provides the "on-source" estimate for the waveform $\hat{h}$, which is the best estimate for the signal given the data. The second analysis is the bootstrap analysis. Firstly, the estimate of the on source signal is removed from both the strain data and the whitened data, to obtain an estimate of the residuals and in both the strain domain and the whitened domain. The residuals are then used to generate the AR-Sieve Bootstrap or the Block Bootstrap.

AR-Sieve Bootstrap

For the AR-Sieve bootstrap, having obtained the residuals for the data and the relative estimate for the noise

$$\epsilon(t) = d(t) - \hat{h}(t) \tag{7.20}$$

we assume the residuals to be distributed as a wide sense stationary normal process, characterized by a PDF

$$\epsilon(f) \sim \mathcal{N}(0, \hat{S}_\epsilon(f)) \tag{7.21}$$

which is fully determined by the knowledge of the power spectral density $\hat{S}_\epsilon(f)$ for the residuals. The power spectral density is estimated by the use of MESA on the colored residuals. Having estimated the PSD, hundreds of thousands of seconds of noise are generated by

sampling the distribution fitted to the data. The reconstructed waveform is then injected as-is in the generated noise, and the pycWB analysis is performed on these data, to obtain the bootstrap replicates for the reconstructed waveform.

Non-parametric Bootstrap

For the non-parametric bootstrap, we start from the whitened residuals, which are an estimate for the error in the data, and we assume them to be i.i.d. random variables drawn from some unknown distribution. The strength of this approach is that it does not make any assumption about the underlying distribution of the data, and it can be used to build confidence intervals even when the distribution of the bootstrap replicates is not normal. The bootstrap samples for the error are obtained by resampling with replacement from the whitened residuals. In this way we obtain N replicates for the noise process and, again, the waveform is injected as is in this new noise realization, and the pycWB analysis is performed to obtain the bootstrap replicates for the reconstructed waveform.

Having obtained the bootstrap, reconstructed replicates, the waveforms are aligned in time with the onsource estimate and the confidence intervals are built by using the percentile method.

7.4.1 Test Case Example

To compare the two methods, we work on a signal injected in simulated gaussian noise at 4 different distances, to check a possible dependence on the SNR of the signal. We inject a BBH signal with the following parameters:

mass1 ($M_\odot$)	mass2 ($M_\odot$)	spin1z	spin2z	RA (deg)	DEC (deg)
33.6	32.2	0	0	174.1	44.3

The resulting values of SNR in the HL network are 19, 28, 71 and 114, corresponding to injected distances of 1.5Gpc, 1Gpc, 500Mpc and 250 Mpc respectively. At first, we investigate the residuals of the on-source analysis, to check if they are independent and normally distributed

in Figure 7.1 we can see the QQ plots for the residuals of the on-source analysis for the 4 different injections, together with the p-values from the Anderson-Darling test for normality. All the residuals appear to be normally distributed around the bulk of the distribution, with some deviation in the tails of the distribution. The tails deviation do not show any strong anomaly across the plots. The p-values are mostly consistent with the normality assumption. Under this assumption, both the parametric and non parametric bootstrap can be used.

The results for different waveform for the two methods are reported in Figure 7.2a for the parametric Bootstrap and figure 7.2b for the non parametric bootstrap applied on the residuals. For representation purposes, the signal is subtracted from the band, so that the confidence intervals should be centered around zero. In both figure, the purple line represents the bootstrap confidence interval, while the green one represents the confidence interval obtained by re-injecting the same signal over and over across stationary data, which should provide a benchmark for the expected variability of the reconstructed waveform, and is the same in both panels.
The results are only reported for the L interferometer, but they are consistent with what

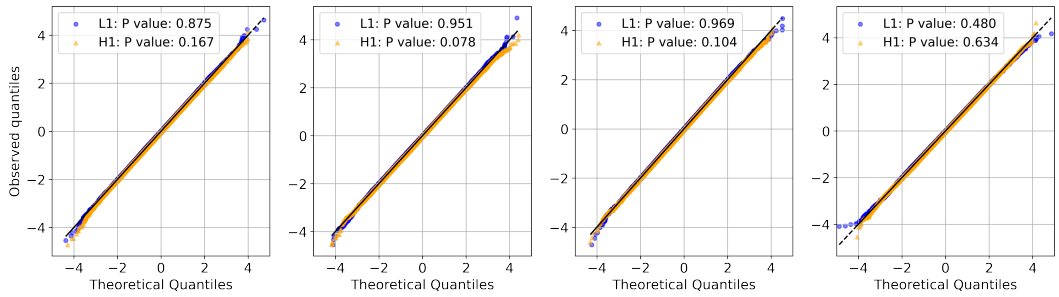
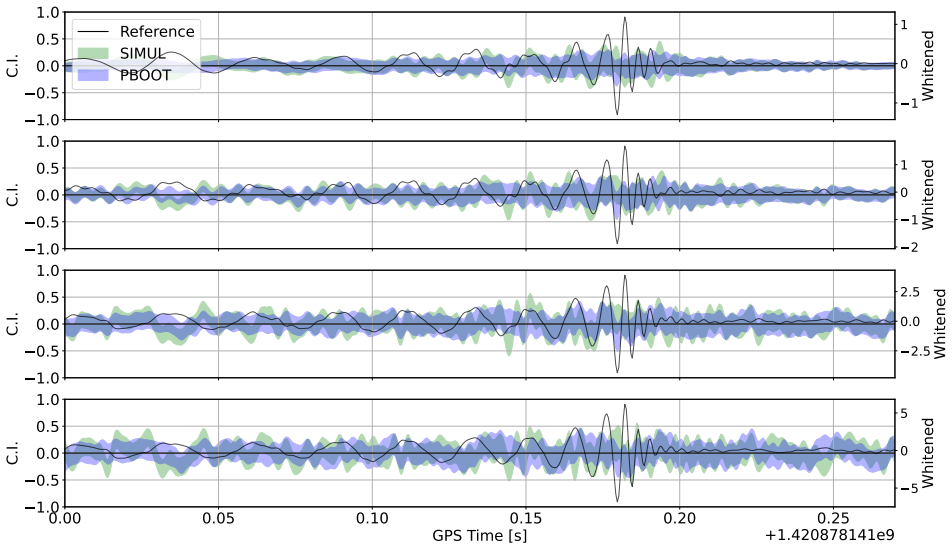
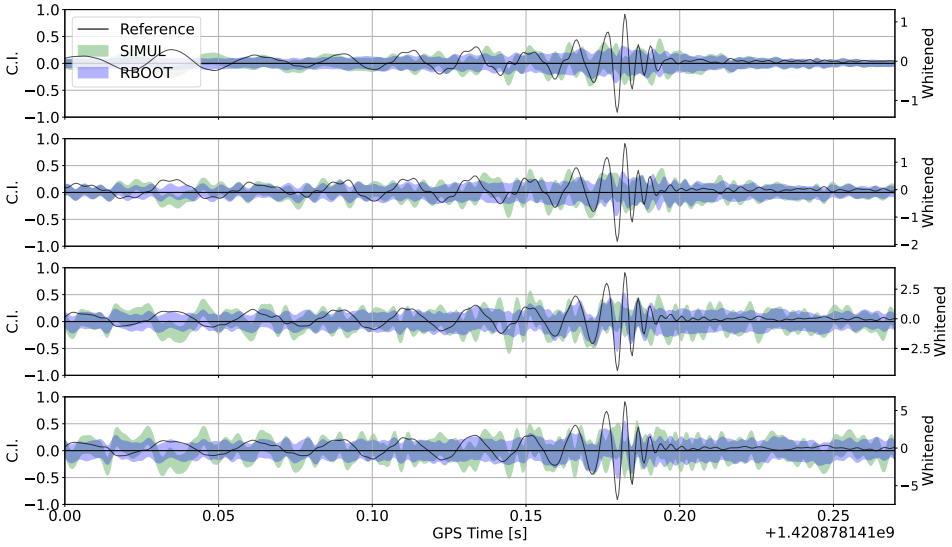


Figure 7.1: QQ plots and p-values from Anderson test for the residuals of the injections. The x axis represents the quantiles that one would expect form a normal distribution, while the y axis displays the empirical quantiles of the observed residuals. From left to right, the 250Mpc, 500Mpc, 1GPc and 1.5GPc injections are reported.

is obtained for the H interferometer. As we can see, in almost all the time domain, the width and the shape of the confidence intervals obtained with the two bootstrap methods provide a good approximation of the confidence intervals obtained re-injecting the signal over and over. The width of the uncertainty band appear to be approximately constant across the time domain, with an exception around the merger times of the waveform, where the confidence intervals are wider (although the percentage error is smaller). At high SNR, the three method display small differences.
The off-source experiment shows a confidence interval whose envelope follows, during the merger time, the envelop of the signal itself. The shape points toward a slight under-estimate of the local signal power, with the confidence shifted towards the opposite side of the zero line with respect to the signal. A similar behavior is observed for the non-parametric signal, but with an opposite direction: the confidence interval follows the oscillation of the signal on the same side of the zero line, which points towards a slight over-estimate of the local signal power. For the parametric bootstrap, no similar behavior is observed, but the confidence interval displays some oscillation at an constant frequency of 20Hz, that may be originating from the PSD estimated in the residuals.
The differences among these methods can be explained by the different procedure used to generate each of the result. The off-source experiment relies on the re-injection of the simulated CBC signal over time. On the other hand, both bootstrap techniques re-inject the on-source reconstructed estimate. If there is a bias in the on-source estimate, it will be reflected in the bootstrap replicates. Finally, the parametric bootstrap injects the signal in stationary, colored noise, and the assumptions made on the PSD filter cannot hold exactly. These different procedure could be the origin of the differences observed in the results.
To compare quantitatively the two methods against the benchmark we rely on two different global metrics: the ratio between the width of the confidence interval for the bootstrap methods and the width of the confidence interval for the off-source experiment, and the difference between the median of the bootstrap replicates and the median of the off-source experiment normalized by the width of the confidence interval for the off-source experiment. The results are reported in Table 7.1 with their mean value and standard deviations. Both the metric provide similar results in term of width of the confidence interval and median of the bootstrap replicates, compared to the off-source experiment. At this level, there is no clear advantage of one bootstrap method over the other, since they both provide a good approximation of the confidence intervals of the off-source experiment, each with its own



(a) Parametric bootstrap confidence intervals for the whitened waveform reconstructed on the L detector. The reported intervals are the 95% around the median.



(b) bootstrap confidence intervals for the whitened waveform reconstructed on the L detector. The reported intervals are the 95% around the median.

Figure 7.2: Bootstrap confidence intervals for the reconstructed waveform. The top panel shows the intervals obtained with the parametric bootstrap, while the bottom panel shows the intervals obtained with the non-parametric bootstrap. In both cases, the reconstructed waveform is compared with the benchmark obtained by repeatedly reinjecting the same signal into stationary noise, represented by the green belt.

Dist. (Mpc)	Method	Width ratio	Median difference
250	Param.	1.0 ± 0.1	0.0 ± 0.3
	Non Param.	1.0 ± 0.1	0.0 ± 0.3
500	Param	1.05 ± 0.06	0.0 ± 0.3
	Non Param.	1.02 ± 0.07	0.0 ± 0.3
1000	Param.	1.04 ± 0.04	0.0 ± 0.3
	Non Param.	0.98 ± 0.05	0.0 ± 0.3
1500	Param.	1.04 ± 0.06	0.0 ± 0.4
	Non Param.	0.94 ± 0.04	0.0 ± 0.3

Table 7.1: Metrics for the comparison between the two bootstrap methods and the off-source experiment. The width ratio is the ratio between the width of the confidence interval for the bootstrap methods and the width of the confidence interval from the off-source experiment.

peculiarities.
Despite the non-parametric bootstrap being model independent, reflecting the principle of burst searches, its implementation in the pipeline requires a more significant change in the configuration and an ad-hoc analysis workflow. On the other hand, the parametric version can be implemented with ease inside the pycWB workflow, which is already compatible with the injection of signals in stationary noise. For this reason, we decide to implement the parametric bootstrap in pycWB and use it for the rest of this thesis.

7.5 Conclusions

In this chapter we introduced the bootstrap method and how it can be used to build confidence intervals for the reconstructed waveform obtained with pycWB. Both the parametric and non-parametric approaches where implemented and compared in a test case by re-injecting the very same signal several times, on simulated stationary gaussian noise. The analysis on the whitened residuals of the on-source analysis do not display any significant deviation from the data being normally distributed as expected. The two bootstrap methods provide slightly different results that could be related to the different procedures used to generate the bootstrap replicates, but they both provide a good approximation of the confidence intervals obtained by re-injecting the same signal several times. It is worth noticing that, since the non parametric bootstrap is performed on the whitened domain, the data conditioning step is not applied in this case, which could create some differences in the results.
None of the two methods exhibit particular deviations. The parametric bootstrap can be implemented more naturally in pycWB, whose workflow is easily compatible with this implementation. On the contrary, the non-parametric bootstrap would requires a more significant change in the configuration of the analysis and the data flow. For this reason, since there are no significant differences between the two methods, we decide to implement the parametric bootstrap and use it in the rest of this thesis.

Chapter 8

Case Study: GW250114

8.1 Introduction

This last section provides an application of the developments presented in the previous chapters. Among all the possible signals, we decided to focus on the special event GW250114[3], which is a highly symmetric binary black hole merger, with a total mass of $70M_{\odot}$ and a mass ratio of 1 at a distance of 400Mpc. This event is interesting for several reasons. First of all, the signal is very loud, with a network SNR of 73.5, and has been easily characterized by the matched filter pipelines, giving us a good benchmark to test what we implemented so far. Starting from the pycWB coherent analysis of the LH data as described(chapter 6) and the bootstrap technique (chapter 7), we discuss here a new method to extract the information from the waveform posterior samples. Though the methodology is very general, we discuss it here in the context of a real signal, because its impact may well be case dependent. This methodology is based on building a "Network Covariance Matrix" for the reconstructed waveforms, that makes us measure the actual degrees of freedom that our network analysis is using. This was not previously implemented in the context of burst searches and it is an original contribution which enables a deeper understanding of the intrinsic correlations in the reconstructed waveforms as well as the development of more powerful statistical tests of consistency with signal models.
In particular, knowledge of the Network Covariance Matrix allows defining a "Mahalanobis" distance [86] between reconstructed and modeled waveforms, which can be used in "decorrelated residuals" tests. We discuss the advantages with respect to tests based on the "overlap" of waveforms. In addition, we discuss applications to the inference on the parameters of the signal model, including a test case on deviations from GR hypothesis. We show this with two simple applications on GW250114, to try and make inference on the signal parameters, which has never been done before, and trying to see if our reconstruction is actually able to discriminate between a variation of the parameter space of the signal under GR or any deviation from the GR hypothesis.

8.2 Building a Network Covariance Matrix

As we said in the introduction, the first step of the new method is building a covariance matrix for the network under consideration. In the case of GW250114, the observation was

performed by the two LIGO detectors. Let's define the on-source estimate of the waveform as $\hat{h}(t)$, and h_i^* the bootstrap reconstructions, with $i = 1, \ldots, N_B$. In our case, $N_B \sim 2 \times 10^4$ parametric bootstrap replicates.
The covariance matrix can be defined per each single detector D in the network as

$$C_D(t_i, t_j) = \frac{1}{N_B - 1} \sum_{k=1}^{N_B} (h^*_{D,k}(t_i) - \bar{h}_D(t_i))(h^*_{D,k}(t_j) - \bar{h}_D(t_j)) \tag{8.1}$$

where $\bar{h}_D(t)$ is the ensemble mean of the bootstrap reconstructions at each time sample. The dimensions of the covariance matrix are $N_t^D \times N_t^D$, where N_t^D is the number of time samples of the reconstruction in the detector under consideration.
The covariance matrix has a non trivial structure, and contains all the information about the statistical error of our reconstruction. In particular, the diagonal elements give the variance of the reconstruction at each time, and can be used to build the confidence intervals as we did in the previous chapters. The off-diagonal elements give the covariance between the reconstruction at different times, and can be used to understand the statistical correlations in the reconstructed waveforms. By construction the covariance matrix is symmetric and positive-definite, but it can be singular, since the reconstruction of the waveform can be correlated among different time samples. This is indeed a common case, as we will show, and it means that the total numbers of the degrees of freedom of the reconstruction is smaller than the total number of time samples. Mathematically, this statement implies that the covariance matrix has a rank smaller than its dimension, i.e.

$$Rank(C_D) = R_{C_D} \leq N_t^D \tag{8.2}$$

The matrix C_D can be diagonalized once we compute the matrix of its eigenvectors V, and its diagonal representation is

$$\Lambda_D = V^T C_D V = \text{diag}(\lambda_1^D, \ldots, \lambda^D_{R_{C_D}}, 0, \ldots, 0) \tag{8.3}$$

with $\lambda_i^D > 0$ for $i = 1, \ldots, R_{C_D}$. The total number of non-zero eigenvalues is the real dimensionality of the problem, i.e. the rank of the covariance matrix. The eigenvectors give us the directions of the space where the reconstruction is not degenerate, while the eigenvalues give us the variance of the reconstruction along those directions. In the following, we focus on the case of the HL network, since at present it is the most sensitive pair of detectors, but the same construction can be implemented for a larger number of detectors.
As a first step, we define the "network reconstruction" as a vector obtained by concatenating the reconstruction of different detectors, i.e.

$$h(t) = [h_L(t_L), h_H(t_H)] = h_L \oplus h_H \tag{8.4}$$

the two reconstruction can have a different number of time samples, namely N_t^L and N_t^H, so that $N_t = N_t^L + N_t^H$. The covariance matrix is defined as in equation 8.1, but with the network reconstruction instead of the single detector reconstruction.
The covariance matrix is then a $N_t \times N_t$ matrix, which contains all the information about the statistical error of our reconstruction for both the single detectors and their pairwise combinations. The covariance matrix has a clear block structure

$$C = \begin{bmatrix} C_{LL} & C_{LH} \\ C_{HL} & C_{HH} \end{bmatrix} \tag{8.5}$$

with the diagonal blocks containing the covariance of the reconstruction in each detector, while the off-diagonal blocks contain the covariance between the reconstruction in different detectors and $C_{HL} = C_{LH}^T$. The total Rank of this matrix is $R_C < N_t$, and gives us the real dimensionality of the problem, taking into account both the correlation of the reconstruction in a single detector and the correlation between the reconstruction in different detectors. The eigenvectors and eigenvalues of this matrix gives us the directions and the variance of the reconstruction in the network space, which is a much more complex and interesting space than the single detector space. When diagonalized, the covariance matrix can be written as

$$\Lambda = V^T C V = \mathrm{diag}(\lambda_1, \ldots, \lambda_{R_C}, 0, \ldots, 0). \tag{8.6}$$

The structure of C_{LH} determines the relationship between the reconstruction of the single detectors in the network and the network reconstruction. In the case of total absence of correlation between the reconstruction in different detectors[1], meaning $C_{LH} = 0$, the covariance matrix is be block diagonal, and the network reconstruction is a simple concatenation of the single detector reconstruction, with the same eigenvectors and eigenvalues, for which $R_c = R_{C_L} + R_{C_H}$

$$\Lambda = \begin{bmatrix} \Lambda_L & 0 \\ 0 & \Lambda_H \end{bmatrix} = \lambda^L \oplus \lambda^H \tag{8.7}$$

This problem would reduce to the direct sum of the single detector reconstruction, and the network reconstruction would not give us any more information than the single detector reconstruction. In our context, there will be some correlation between the reconstruction in different detectors, meaning $C_{LH} \neq 0$, the covariance matrix will not be block diagonal, and the network reconstruction cannot be reduced to the direct sum of the single detector reconstruction, with different eigenvectors and eigenvalues, implying that $R_c < R_{C_L} + R_{C_H}$, i.e. the network of detector is actually further decreasing the total number of degrees of freedom of the problem than what we would have by looking at the union of the single detector analysis, meaning $\Lambda \neq \Lambda_L \oplus \Lambda_H$. If the case of perfectly correlated and identical detectors, meaning $C_{LH} = C_{LL} = C_{HH}$, the covariance matrix takes the form:

$$C = \begin{bmatrix} C_{LL} & C_{LL} \\ C_{LL} & C_{LL} \end{bmatrix} \tag{8.8}$$

and $R_C = R_{C_L} = R_{C_H}$. Having defined the two extreme cases, we find that the following inequality holds:

$$\max(R_{C_L}, R_{C_H}) \leq R_C \leq R_{C_L} + R_{C_H}. \tag{8.9}$$

In our scenario, since the H and L detectors share similar directional and spectral sensitivities, we can expect that the correlation between their reconstructed waveforms is far from negligible. The difference $R_C - R_{C_L}$ can be seen as a representation of information content brought by the joint analysis of the detectors with respect to the single detector reconstruction. Let's see how this applies to the case of GW250114.

8.2.1 The Network Covariance Matrix for GW250114

We apply the methodology described in the previous section to the case of GW250114, building the covariance matrix for the LH network with the parametric bootstrap, with a total number of bootstrap replicates of about $N_B = 2 \times 10^4$. In the top Panel of Figure 8.1 we show the on-source reconstruction for GW250114, together with the $\pm 3\sigma$ confidence interval

[1] This is unrealistic in the context of candidates identified by a coherent network analysis such as pycWB.

around the signal, obtained with the parametric bootstrap using the standard deviation as a measure of uncertainty. In the bottom panel, ± the data are plotted against the $\pm 1\sigma$ confidence interval, now centered around zero, to show the shape of the confidence interval. From the picture it emerges clearly that the confidence interval is not constant in time, and that the absolute uncertainty increases monotonically with the the amplitude of the signal. At the merger, the envelope of the confidence interval is modulated at a frequency approximately twice that of the signal, which is a consequence of the fact that the main source of error in our reconstruction is monotonically increasing with the absolute amplitude of the signal.

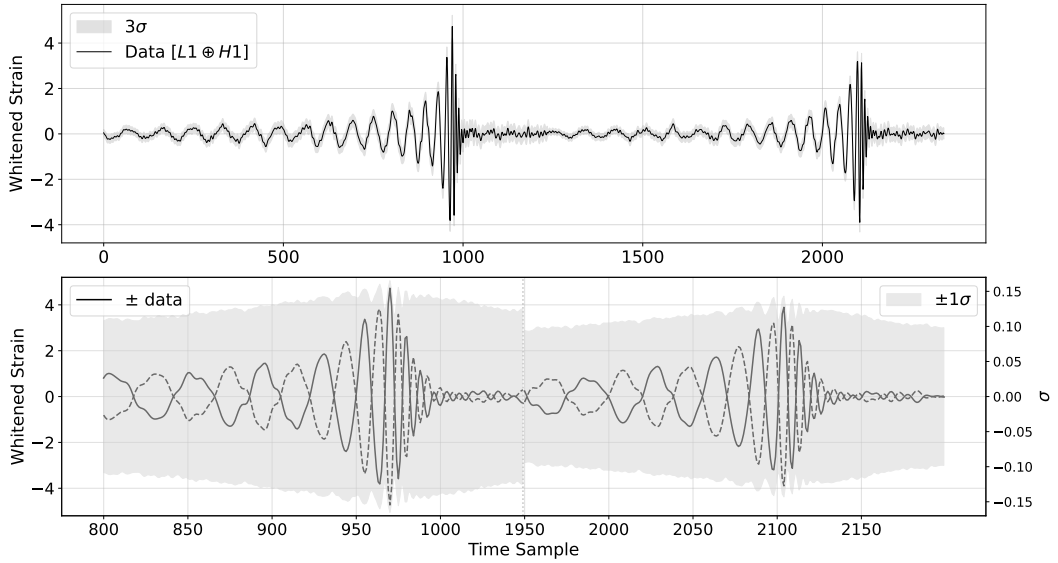


Figure 8.1: Top panel: On-source reconstruction for GW250114, with the $\pm 3\sigma$ confidence interval around the signal obtained with the parametric bootstrap. The confidence intervals are directly obtained from the square root of the diagonal elements of the network covariance matrix. Bottom panel: on-source reconstruction (gray solid line) and minus the on-source reconstruction (dashed gray line) for GW250114 and $\pm 1\sigma$ relative uncertainty. The confidence intervals displays a time modulation which grows monotonically with the amplitude of the signal.

To understand the degrees of freedom of the problem, we diagonalize the covariance matrix to obtain the total set of the eigenvalues and eigenvectors. We compare four analyses of the reconstructed waveform provided by pycWB on the HL network:

- the analysis exploiting a single detector (i.e considering only C_{LL} or C_{HH}, separately), which would be optimal in the extreme case of fully degenerate responses from the two detectors;
- the analysis considering the direct sum of the responses L $\oplus$ H, which would be suitable in the opposite extreme case of independent waveform information from H and L, as if the two detectors were measuring orthogonal polarization states of the signal;
- the analysis taking into account the structure of the correlation between the waveforms reconstructed by H and L contained in C_{LH}.

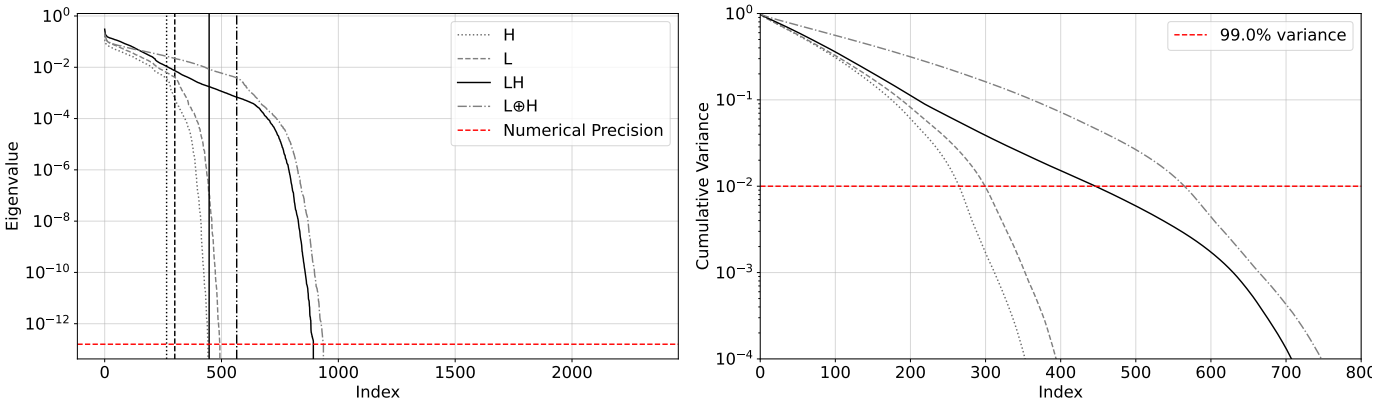


Figure 8.2: Left panel: Eigenvalues of the Network covariance matrix for the reconstructed waveforms of the single detectors of the network, H and L, reported as dotted and dashed lines respectively, the LH correlated network (solid line) and the direct sum of the detectors taken independently $L \oplus H$ (dashed dotted line) The higher the value for the eigenvalue, the higher the variance of the reconstruction along the corresponding eigenvector. Right panel: The cumulative variance as a function of the total number of eigenvectors. The dimension of the matrix is $N_t = 2339 \times 2339$, with $N_t^L = 1235$ and $N_t^H = 1104$.

	H	L	LH	L⊕H
50.0	61	62	69	121
90.0	175	188	210	363
95.0	209	230	275	439
99.0	265	300	447	565

Table 8.1: Number of eigenvalues required to explain a given percentage of variance for L, H, their union, and their sorted combination.

The spectrum of eigenvalues is shown on the left panel of Fig. 8.2 for the four different reconstruction procedures. The spectrum ranges a large number of orders of magnitude, from $\mathcal{O}(10^{-1})$ to $\mathcal{O}(10^{-20})$ and lower, meaning that a large number of these eigenvalues are actually compatible with zero within numerical precision. Together with the spectrum, for each case we also report in as a vertical line the total number of eigenvalues required to explain the 99 percent of the total variance, defined as the cumulative sum of the eigenvalues, divided by their total sum. The numbers at some fixed thresholds reported in Table 8.1.
The number of eigenvalues required to explain a high fraction of variance is a good proxy for the number of degrees of freedom of the problem and the rank of the covariance matrix. The right panel of Figure 8.2 shows the cumulative variance as a function of the eigenvalues number for L and H taken separately and their network reconstruction, with or without correlation.
In the case of L and L $\oplus$ H, the 99% of the variance seem to be a good proxy for the rank of the matrix, since after this value the eigenvalues start dropping faster towards zero, so that it can be taken as a threshold. Looking at the eigen-spectrum for LH is different, with an exponential decay at a constant rate up until the 99.9% threshold. This threshold is extremely high considered that we used 10^4 bootstrap replicates, For this reason, since for

both L and LH the 99% of the variance appears to be a good estimate for the number of degrees of freedom, we use the same threshold for the LH cases, which means that we are considering 447 degrees of freedom.
It is interesting to compare the number of degrees of freedom for L, H and their combinations as a function of the variance threshold, as shown in table 8.1. First of all, it is worth noticing that, as it should be, the number of degrees of freedom for L $\oplus$ H is approximately the sum of the number of degrees of freedom for L and H. It is expected since in this way we are treating the reconstruction in each detector as completely independent. In some cases there are some small deviations, compatible with the discretization of the data.
Also notice that, for every threshold, the number of degrees of freedom for H is systematically smaller than the number of degrees of freedom for L, which is expected since the SNR in H is smaller than the SNR in L, hence L is is reconstructing more signal-related information. So we can use L as a "reference" for the discussion.

At 50% and 90% , the two detectors appear to be highly correlated, with a number of degrees of freedom for LH being only slightly larger than the total number of degrees of freedom from L alone. In other words, H adds a non-negligible but anyhow small advantage in waveform reconstruction of the main signal features. This can be explained by the fact that the two detectors are close, with directional and spectral sensitivities, so that they are sensitive to the same signal polarization (f_1 in the dominant polarization frame). There correlation decreases significantly at the 99% threshold, where 147 more degrees of freedom are needed to explain the variance for LH with respect to L, meaning that the combination of the two detectors is actually bringing a significant amount of information to the waveform reconstruction for what concerns weaker signal features. In general, we can see that for this event, we are closer to the case of highly correlated detector responses than to the ideal case of a network able to measure both GW polarization components. At the same time, since pycWB reconstruction based on a coherent analysis in the network of the detectors and uses the correlated energy information among the detectors in the network to reconstruct the signals, a large correlation can also be explained by our reconstruction method.

8.2.2 The independent degrees of freedom of the waveform reconstruction

It is interesting and informative to to discuss the eigenvalues of the network covariance matrix, but also at the corresponding eigenvectors, which give us the orthogonal directions of the errors in the reconstruction of the waveforms. We focus and compare the first few eigenvectors for the network covariance matrix with both the onsource-reconstruction and the first eigenvectors of the reconstructions separately provided by L and H waveforms. Let's call the eigenvectors as $\vec{v}_i^D$ with i the number of the correspondent eigenvalue and D the detectors. It is more interesting to see how the eigenvectors of the network are built and to see if any of them exhibit any clear structure in terms of the single detector eigenvectors and how the correlation structure emerges from the combination of the two detectors.

In the top panel of Figure 8.4, the first eigenvector is plotted against the on-source reconstruction. The two curves are almost coincident in frequency evolution, and this first source of error lays on the same (mathematical) direction of the signal: the error in the reconstruction is proportional to the signal itself, therefore it represents the error on the estimate of the amplitude of the signal. This is compatible with what we observed in the confidence interval of the on-source reconstruction shown in Figure 8.1. The central panel

shows the first eigenvector of the network and the direct sum of the single detector eigenvectors. Both eigenvectors are normalized to have unit norm. For the first direction, the network eigenvector is almost aligned with the direct sum of the single detector eigenvectors, indicating that the relevant reconstruction is the same whether considering the single detectors separately or the network as a whole. This suggests that correlation is not relevant for this reconstruction. Nevertheless, it is worth noting that the ratio of squared amplitudes (defined as the sum of the squared components) differs between the two cases. When the direct sum is considered, the two components contribute equally to the total amplitude by construction. When correlation is included, L contributes 65% of the total amplitude, while H contributes the remaining 35%. The difference in sign between the two components of the

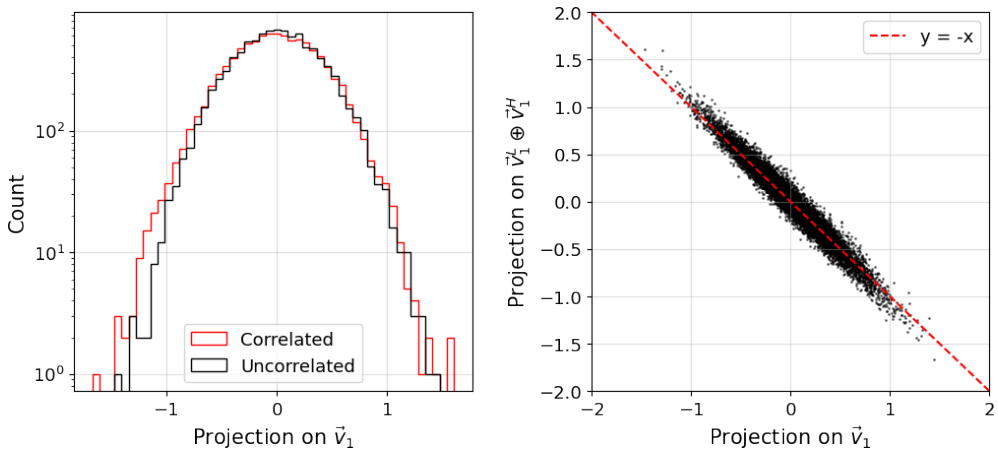


Figure 8.3: Comparison of the projection onto the first eigenvector for the network and the first eigenvector for the direct sum of the detectors taken separately. The left panel shows an histogram of the values for the projections, while the right panel shows a scatter plot of the projections. The two are almost anti-aligned, and the angle between the two eigenvectors is found to be $\sim -170°$

network eigenvector and the direct sum does not carry any particular meaning, since the eigenvectors are defined up to a sign. To show this, in Figure 8.3 we show the projection of the residuals of the bootstrap reconstructions onto the first eigenvector for both the network and the direct sum. On the left plot the histogram of the projection for the network and the direct sum are shown, while on the right plot shows a scatter plot for the two cases. For both the Network and the direct sum, the projection of the residuals onto the eigenvector basis is distributed symmetrically around zero, with comparable magnitudes, so that the different sign of the components of the eigenvector for the two cases does not have a particular meaning.
The last panel of the figure shows the second eigenvector, together with the direct sum of the second eigenvectors of the single detectors. First, we note that it is difficult to identify any waveform-related physical meaning in this eigenvector. While the first component is very close to the L eigenvector, which is pointing to a low frequency component of plausible noise origin, the second component is quite different from the H eigenvector. None of the other eigenvectors from H up to order 12 showed any similarity with this second component, so it is not possible to describe it as a combination of single detector eigenvectors. Therefore, it must carry a different type of meaning. From the third eigenvector onward, the structure

of the eigenvectors becomes increasingly difficult to interpret, and we do not discuss them further.
The interpretability of the eigenvector is a powerful tool, and can highlight the sources of error in our reconstruction method. This was actually the case for the firstly obtained results of this investigation: the NCM pointed at a systematic error in the alignment of the waveforms, allowing us to correct the code to take that error into account. The preliminary results together with our first interpretation are reported in Appendix F

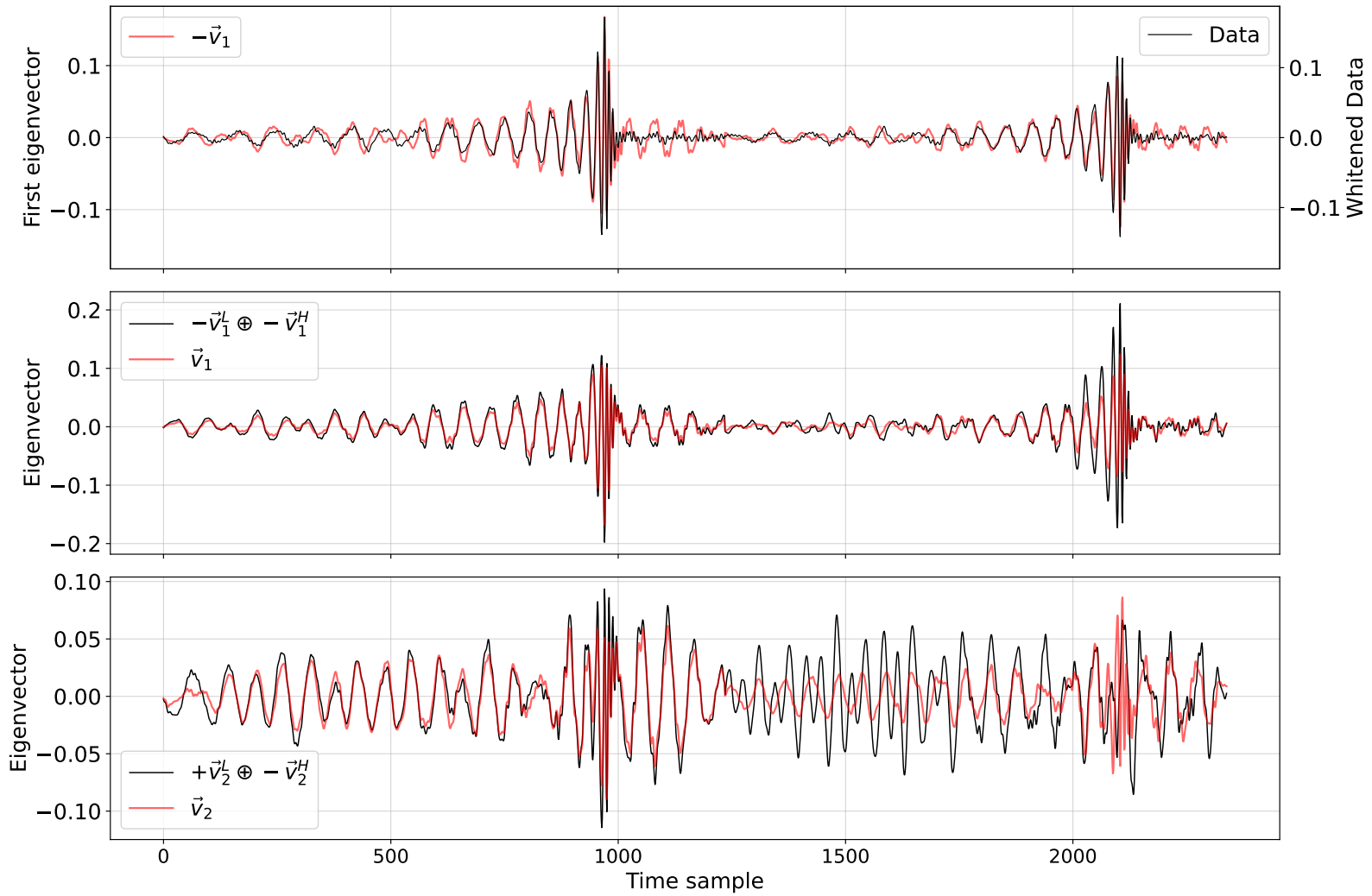


Figure 8.4: First two eigenvectors of the network covariance matrix (red curves), together with the data (top panel) and direct sum of the first two eigenvectors of the single detector reconstruction (central and bottom panels). The network eigenvectors are normalized to have unit norm, while the single detector eigenvectors are normalized to have unit norm in their respective subspace.

8.3 Making inference with the NCM

The network covariance matrix (NCM) can enable statistically meaningful inferences from burst waveform reconstruction. The covariance matrix contains all the information about the statistical error of our reconstruction, so it can be used to build a "Mahalanobis" distance between our best on-source reconstruction and whatever model we want to compare it with, a statistic that gains a lot in discriminating power with respect to the simple overlap used so far as the metric in standard burst searches.
The Mahalanobis distance of a target point h^* from a reference is defined as [86]

$$\mathcal{M}^2(h^*) = (h^* - \bar{h})^T C^{-1} (h^* - \bar{h}). \tag{8.10}$$

where $\bar{h}$ is a reference reconstruction, h^* is a point whose distance from the reference we want quantify and C, the network covariance matrix, weights the various components.

The computation of the Mahalanobis distance is straightforward, but it requires the inversion of the covariance matrix, which, being singular, cannot be inverted in a standard way, meaning that there is no C^{-1} such that $C^{-1}C = I$. This problem can be solved by computing the "pseudo-inverse" of the covariance matrix (or the Moore-Penrose inverse) C^+, which is guaranteed to be unique and always exists, even for singular matrices. In the eigenbasis, the pseudo-inverse of a singular matrix is computed easily as:

$$\Lambda^+ = \mathrm{diag}(\lambda_1^{-1}, \ldots, \lambda_{R_C}^{-1}, 0, \ldots, 0) \tag{8.11}$$

where the inverse of the non-zero eigenvalues is computed, while the zero eigenvalues are left unchanged. In our case, the singular eigenvalues are not exactly zero, so that, as we showed before, we had to define a threshold to decide which eigenvalues are compatible with zero and which are not. In our case, we decided to use the 99% of the variance, and this makes possible to compute the pseudo inverse of the variance of the matrix. The procedure of eigenvalues decomposition, thresholding and pseudo-inversion is equivalent to perform principal component analysis on the problem. With this definition, the Mahalanobis distance is computed as:

$$\mathcal{M}^2(h^*) = (h^* - \bar{h})^T C^+ (h^* - \bar{h}) \tag{8.12}$$

. The Mahalanobis distance function defines a general framework for inference in reconstruction space. Given a reference distribution characterized by some reference point $\bar{h}$ and covariance C. The reference distribution can be built from any hypothesis H by injecting signals consistent with H into noise realizations and running the pycwb, yielding both $\bar{h}$ and C. Depending on the inference goal, this framework supports three distinct use cases.

Parameter estimation. The on-source reconstruction $\hat{h}$ serves as the reference $\bar{h}$, and C is the Network Covariance Matrix encoding the statistical variability of $\hat{h}$ due to noise, estimated from the bootstrap replicates. We sample the parameter space by performing a set of simulated injections, where for each parameter configuration θ_k, the simulated waveform is processed through `pycWB` to yield its corresponding reconstructed waveform vector $h^{(k)}$. One then computes the Mahalanobis distance $\mathcal{M}^2(h^{(k)})$ for each reconstructed injection, building a distribution of distances across the parameter space. The best-fit parameters are identified as those corresponding to the injection that minimizes the distance from the on-source estimate:

$$\hat{\theta} = \theta_{\hat{k}} \quad \text{where} \quad \hat{k} = \arg\min_k \ \mathcal{M}^2\left(h^{(k)}\right) \tag{8.13}$$

Model comparison. Given two hypotheses H_0 and H_1, one builds the two distributions $\{\mathcal{M}^2(h^{(k)}_{H_0})\}$ and $\{\mathcal{M}^2(h^{(k)}_{H_1})\}$ by injecting signals under each hypothesis, where k

indexes the individual injection replicates, and computing $\mathcal{M}^2$ for each reconstructed waveform $h_H^{(k)}$. Here $\bar{h}$ is the on-source reconstruction and C is the Network Covariance Matrix. Comparing the two resulting distributions allows one to assess which hypothesis is more consistent with $\hat{h}$.

Hypothesis testing. Given a hypothesis H_0, one generates $N_{\rm inj}$ reconstructions under H_0 and computes $\mathcal{M}^2$ for each replicate k with respect to the hypothesis mean $\bar{h}_{H_0}$, building the null distribution $\{\mathcal{M}^2(h_{H_0}^{(k)})\}_{k=1}^{N_{\rm rec}}$, with $N_{\rm rec}$ the total number of recovered injections. In this case, $\bar{h}$ is no longer the on-source reconstruction, but the mean over all the reconstructions $h_{H_0}^{(k)}$. C is again the Network Covariance Matrix, but is now built from the reconstructions obtained under the null hypothesis. One then computes the observed statistic

$$\mathcal{M}^2_{obs} = \mathcal{M}^2(\hat{h}) = (\hat{h} - \bar{h}_{H_0})^\top C^+ (\hat{h} - \bar{h}_{H_0}) \tag{8.14}$$

measuring how far the on-source reconstruction sits from the hypothesis mean. The empirical p-value is then obtained as the fraction of recovered injected waveforms whose distance exceeds the observed one:

$$p = \frac{1}{N_{\rm rec}} \sum_{k=1}^{N_{\rm rec}} \mathbf{1}\Big[\mathcal{M}^2\left(h_{H_0}^{(k)}\right) \geq \mathcal{M}^2_{obs}\Big] \tag{8.15}$$

8.3.1 Comparing with GR templates

In the previous section, we demonstrated how the distribution of the distance between our estimate and a model can be used to generate an empirical distribution for the distance between the pycWB on-source estimate for the signal and a hypothesis H_0 that describes the signals in terms of a model and a set of parameters. In this case, the parameters are those provided by the CBC parameter estimation [3], and the model is the approximant used to generate the waveform. For GW250114, the re-injection is performed using the parameters provided by Bilby [26] with the "IMRPhenomXPHM-SpinTaylor" [97] approximant.

As shown in Figure 8.5, the distribution of the distance under the hypothesis that the CBC signal model be the maximum a posteriori (MAP) vs. a random waveform posterior sample changes drastically, resulting in a completely different distribution, with values for the $\mathcal{M}^2$ variable collapsing towards lower values for the MAP estimate.
It is interesting to notice that, once the full set of posterior samples is considered, together with a large, heavy tail on the right side of the distribution, there is a non-negligible fraction of the samples that have a distance smaller than the MAP estimate. We already have evidence that there are a set of parameters that are closer to our reconstruction than the MAP estimate. The larger support of the $\mathcal{M}^2$ distribution indicates that our reconstruction is sensitive to at least a sub-set of parameters. In a sense, we can perform "parameter estimation" using the pycWB reconstruction and the NCM by identifying the set of parameters that minimizes the Mahalanobis distance between the on-source reconstruction and the model.
The procedure is as follows:

- We select a set of posterior samples obtained from the CBC parameter estimation, and for each sample, we generate the corresponding waveform using the same approximant used for the parameter estimation.
- The waveforms are injected into the off-source data and reconstructed with pycWB.

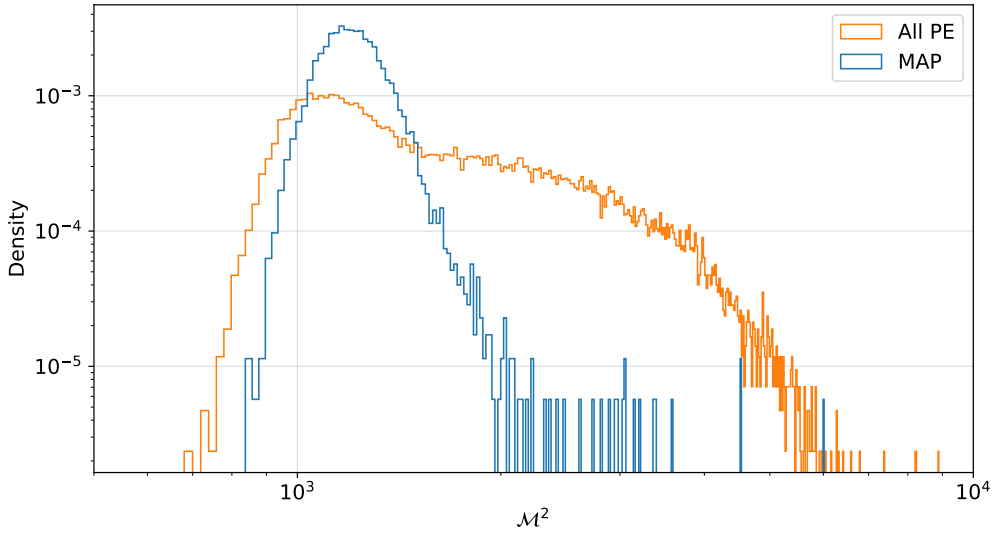


Figure 8.5: Distribution of the Mahalanobis distance between the on-source reconstruction and the model generated with the parameters provided by Bilby [26]. The blue curve shows the distribution of $\mathcal{M}^2$ for the MAP estimate, while the orange curve shows the distribution for the full set of posterior samples.

- We compute the Mahalanobis distance between each of these reconstructions and our on-source reconstruction, using equation 8.12.
- We analyze the distribution of the distance as a function of the parameters, looking for the minimum value, which corresponds to the set of parameters closest to our reconstruction.

In principle, this process could be entirely independent of the CBC samples, since it does not require any information about the morphology of the parameter's space. However, in real scenarios, the parameter space is so huge that the computational cost of an alternative approach would not be justified here. In fact, the CBC samples are based on reliable methods for exploring the parameter space, such as Monte Carlo or Nested Sampling, a resampling of the CBC posterior samples is the most effective way to target a relevant portion of the parameter space.
We focus our attention on a subset of all the parameters for the reconstruction, specifically some of the intrinsic parameters of the binary system, the chirp mass and the effective spin, defined as:

$$\chi_{\text{eff}} = \frac{m_1\vec{\chi}_1 + m_2\vec{\chi}_2}{m_1 + m_2} \cdot \hat{J} \tag{8.16}$$

where m_1 and m_2 are the masses of the two compact objects, $\vec{\chi}_1$ and $\vec{\chi}_2$ are their dimensionless spin vectors, and $\hat{J}^2$ is the unit vector along the orbital angular momentum of the system. We also consider the extrinsic parameters of the system, such as the luminosity distance d_L, the inclination angle ι, and the coalescence phase ϕ. The sky position is represented here in terms of geographical coordinates with the longitude [Long] and latitude [Lat].

[2]By convention the parameters is defined at a reference frequency of 20 Hz.

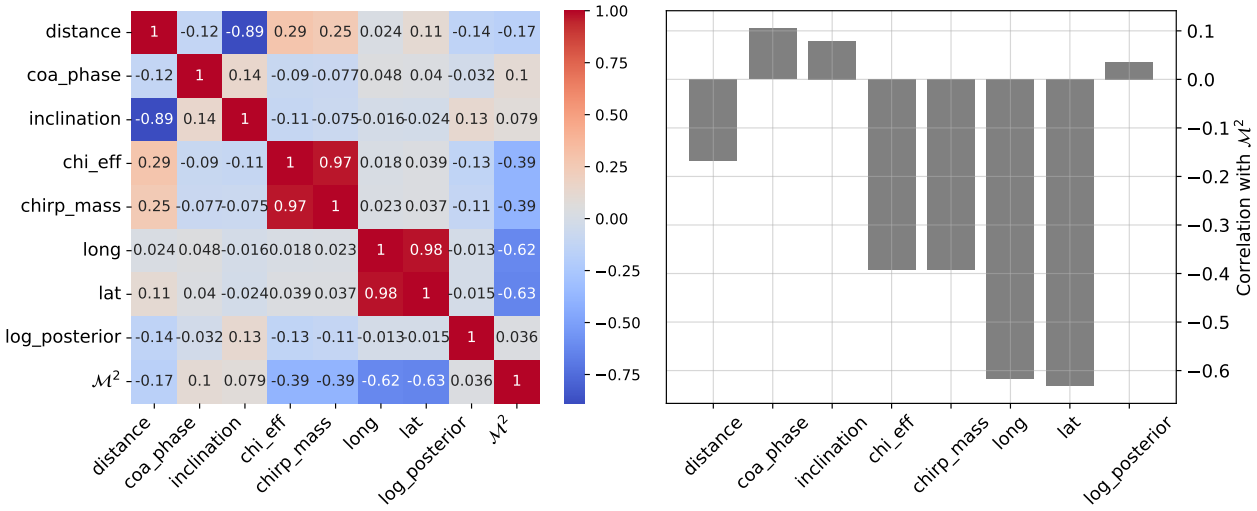


Figure 8.6: Correlation matrix between the selected CBC parameters and the Mahalanobis distance. The parameters considered are: the chirp mass, the effective spin chi_eff, the luminosity distance d_L, the inclination angle, the longitude [Long], and the latitude [Lat].

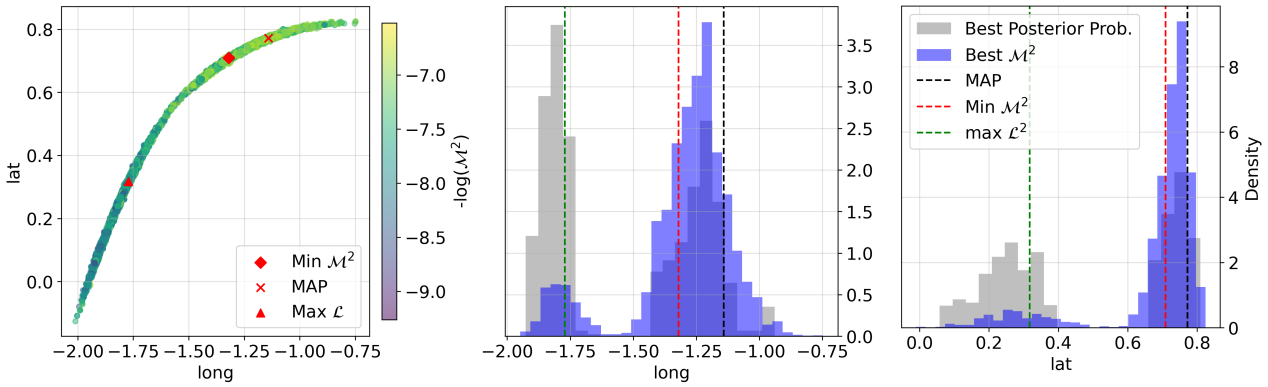


Figure 8.7: Mahalanobis distance as a function of the sky localization parameters. The left panel shows the scatter plot of the two variables (longitude and latitude), which display a clear correlation. The color represents $-log(\mathcal{M}^2)$. The diamond indicates the minimum $\mathcal{M}^2$ point, while the cross represents the MAP estimate for the sky location. The central and right panels represent the marginal distributions for the two variables, conditional to the lowest 10% values of $\mathcal{M}^2$ distance (blue) and the highest 10% values of posterior probability (gray) Both best point estimates are shown in both panels. For comparison, the Maximum Likelihood estimate (MLE) is also shown, reported in the marginal distributions as the green vertical line and the red triangle in the scatter plot.

As a first result, it is interesting to observe how the parameters correlate with each other and which ones are correlated with the Mahalanobis distance. Figure 8.6 shows the correlation matrix of the selected parameters, along with the correlation values with the Mahalanobis distance. The correlation matrix is defined as The correlation matrix, defined as

$$Corr(\theta^a, \theta^b) = \frac{\sum_i (\theta_i^a - \bar{\theta^a})(\theta_i^b - \bar{\theta^b})}{\sqrt{\sum_i (\theta_i^a - \bar{\theta^a})^2 \sum_i (\theta_i^b - \bar{\theta^b})^2}} \tag{8.17}$$

with a,b the indices running over the parameters and i the indices running over the samples. Some of these variables are highly correlated with each other, such as longitude and latitude (which is expected since the estimate lies on a circle of almost constant time delay δt between the detectors) and the chirp mass with the effective spins. The Mahalanobis distance is correlated with some of these parameters and exhibits the strongest correlation, ~ -0.62, with the two sky location parameters. This suggests that we may be able to constrain e.g. the sky location parameters using the pycWB reconstruction.
The CBC posterior samples of the direction are shown in Figure 8.7. The left panel displays the scatter plot of the two correlated angular variables. The color represents the value of $-\log(\mathcal{M}^2)$ distance, indicating that points with the lowest $\mathcal{M}^2$ have the highest color intensity. The correlation of the distance with the sky location is evident. The sky positions corresponding to the minimum $\mathcal{M}^2$ distance and the MAP are represented in the plot with a diamond and a cross, respectively. We observe that the two estimates are not significantly different. The central and right panels of the same figure show the marginal distributions for the two angular variables, conditional to the lowest 10% values of $\mathcal{M}^2$ distance and the highest 10% values of posterior probability. Both best point estimates are shown in both panels as vertical lines.
It is worth noting the difference between the two marginal distributions of the posterior and the $\mathcal{M}^2$ selected points. The highest posterior samples are more sparse and cover a wider range of values for the two variables, displaying a strong bimodality for both longitude and latitude. In contrast, the lowest $\mathcal{M}^2$ samples are concentrated in a smaller range and the bimodality is very weak. Similar distributions for the other CBC parameters are reported in Appendix G. Among all the selected CBC parameters, luminosity distance and inclination angle are the ones showing the highest discrepancy of point estimates between the two methods.

Despite being theoretically interesting, performing CBC parameter estimation with waveform-agnostic burst pipelines is obviously suboptimal, since it requires a set of assumptions, such as the model to be tested (the approximant used) and the set of parameters to be tested (burst methods are are unable to sample the parameter space with a statistically motivated rule). Under these assumptions, it makes little sense to perform full parameter estimation using burst pipelines: since all of these assumptions are better exploited in the context of template-based methods. The methodology developed here could be useful in cases of multi-modal posteriors or Monte Carlo chains that do not easily converge toward the maximum posterior. In addition, the method could also be useful if cae of departures from gaussianity or suspected missing physics in the waveform. In more standard cases, CBC parameter estimation is the best tool to perform parameter estimation, and our methodology is not designed to compete with it.

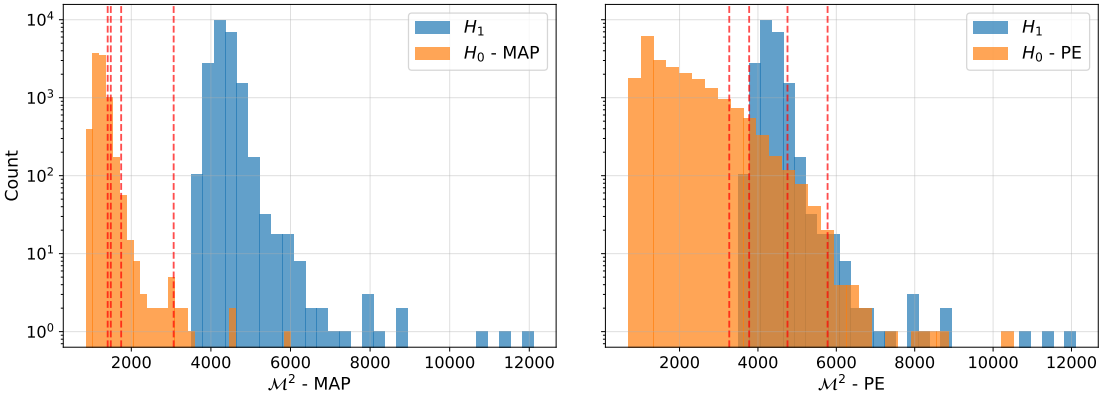


Figure 8.8: Distribution of the Mahalanobis distance for the two models: the GR model with $dchi0 = 0$ (orange) and the non-GR model with $dchi0 \in [0, 7]$ (blue). The left panel shows the two distributions with GW signal parameters fixed to MAP estimates of GW250114 and varying the post-GR parameter in H_1. The plot on the right shows the distribution for the two models with the full set of parameters varying over the posterior samples for H_0, while H_1 is kept the same.

8.3.2 Model Selection: Testing a non-GR waveform

Our methodology can be exploited in model selection between two alternative hypotheses H_0 and H_1, describing the signal under models M_0 and M_1. As an example, we compare a GR waveform model, the GW250114 one used in the previous section, against a non-GR modification of its waveform. Our scope is to show the improvement that our method brings with respect to the methods already established in the context of minimally modeled analyses.

Unfortunately, there are no available waveforms with post-GR models that we can use, but the waveform generator calling LALSimulation has different "post-GR" parameters at different post-Newtonian orders, which can introduce modifications to the waveforms in ways not allowed by standard GR parameters.

In our exercise, we test the effect of the first "post-GR" parameter at 0PN order, called $dchi0$[40][3]. We define the alternative hypotheses as follows:

- H_0: The signal is described by the GR model, with $dchi0 = 0$, and parameters either fixed to the MAP estimates for GW250114 or allowed to vary over the posterior samples.

- H_1: The signal is described by a non-GR model, with $dchi0$ uniformly random in $[0, 7]$, with the GR parameters set to the MAP estimates for GW250114.

The results of the $\mathcal{M}^2$ distributions are shown in Figure 8.8. On the left, we plot the extreme case in which the GR parameters are fixed to the MAP estimate for both models. In this case, the two distributions are almost completely separated, with the non-GR model showing a significant shift towards higher values of the Mahalanobis distance. For any

[3]the $dchi0$ parameter gives the relative deviation from the 0 Post Newtonian coefficient in the GR emission model for the CBC inspiral phase. The best upper limits on this parameter are set by binary pulsar measurements [76].

chosen value of the type I[4] error, the power[5] of the test is very close to 1, meaning that we can always reject the GR model in favor of the non-GR model with such a wide prior on the *dchi*0 parameter. Such a large range of values for *dchi*0 is however inconsistent with observations.

In fact, the upper limits on *dchi*0 from GW modeled analyses of GW250114 are of the order of a few $\times 10^{-2}$ [3] and binary pulsar measurements constrain *dchi*0 at $\sim 10^{-4}$ [76]. Despite the fact that the "dchi0" parameter is sampled in the uniform range starting at 0, the distribution of the alternative hypothesis does not tend to the null hypothesis for small values of the parameter. Checking sample parameters, the minimum value sample for *dchi*0 is $2 \cdot 10^{-4}$, which is two times the pulsar upper limit. Even for such a small value of the parameter, the differences in the waveform are significant enough to shift the distribution of the Mahalanobis distance to higher values, allowing for a clear separation between the two distributions.
The right panel shows a more interesting scenario, where we allow the GR parameters to vary over the whole set of posterior samples, using $\mathcal{M}^2$ as our test statistic. Here, the distributions of the two alternative hypothesis are not fully separated. What is relevant for our model selection purpose is to compare the power of this statistical test with the ones of other two methods based on pycWB and already established. Figure 8.9 reports this comparison of test power versus type I error. The shaded belts represent the 95% interval around the estimated power.

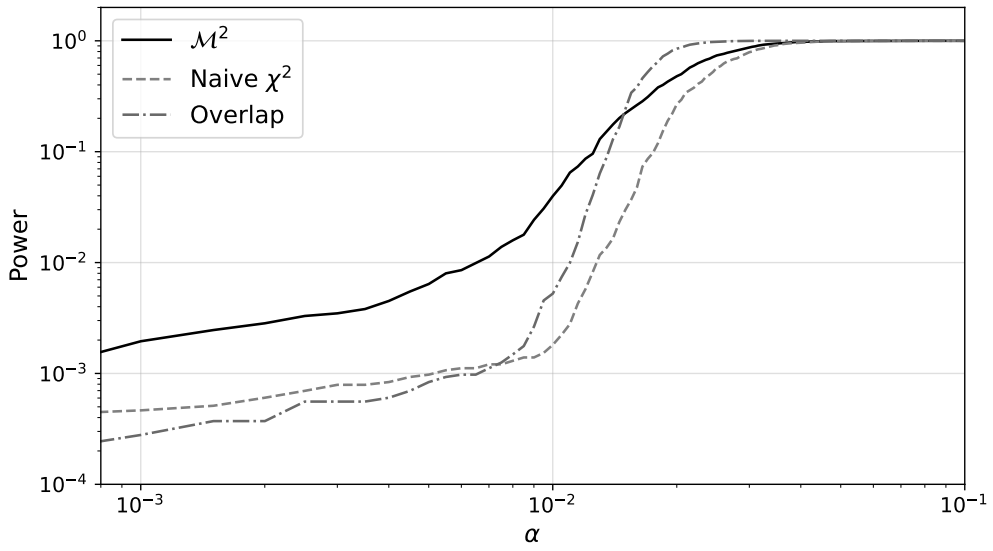


Figure 8.9: Power of the test for different value of the type I error (α). The black line represents the power of the test based on the Mahalanobis distance, while the gray dashed and dash-dotted lines represent the power of the tests based on the Naive χ^2 and the overlap, respectively. The three test procedures use the same pycWB results from one off-source simulation of GW250114 waveform posterior samples. The bands represent the 95% confidence band on the estimate of the power

[4]The probability of rejecting the null hypothesis when it is true.
[5]The power of a statistical test is the probability that the test will reject the null hypothesis when the alternative hypothesis is true.

The pre-existent methods are based on the overlap between the injected and reconstructed waveforms and on a "Naive χ^2" statistic.

The Naive χ^2 is built by summing the square residuals of the time samples, i.e. neglecting all the correlations between them, either within each waveform as reconstructed from a single detector and across the waveforms from different detectors. The overlap is the standard test statistic actually used in burst searches to compare waveforms and in particular to provide the test of consistency between burst reconstructions of GW transients and the CBC PE waveform posterior samples [41].

As it is evident from Figure 8.9, the $\mathcal{M}^2$ test is more powerful than the other two statistics for a relevant range of type I error. The improvement in terms of power can reach up a factor 20 at $\alpha \leq 10^{-2}$, reaching an improvement in the power of a factor $\sim 5, 10$ for the lower values of α. Consistently with past investigations, the results show that the overlap performs overall slightly better than the naive χ^2.

This exploratory study is a clear demonstration of the importance of accounting for the correlation structure of the reconstructed waveforms, and of the improvement in power of the test achievable by NCM. More investigations are needed to test the method to a wide variety of model selection cases, that this work have no time to address. In particular, these should include non-CBC signals, where competing models may predict pronounced differences in the predicted signals. Nevertheless, we can conclude that NCM provides the framework to perform better model selection analyses in the context of minimally modeled searches for GW transients.

8.4 Hypothesis Testing: NCM for improved waveform consistency test

What can be more interesting at this level is using it to perform pure hypothesis testing against a null hypothesis H_0 without an alternative hypothesis $H1$. This is what is usually done in the context of "waveform consistency test" for CBC signals, in which some metric is used to build a distribution of the expected differences between burst reconstructions and the CBC signal models. The on-source estimates from the burst analysis and from the CBC signal model are then compared to assess an "observed" metric value and a p-values is assigned to the observation.

The waveform consistency tests [41] performed in standard LVK analysis uses the overlap as the metric for distance between the two waveforms. The H_0 distribution of the overlap is obtained by performing several injections of the GR templates on off-source data and comparing the reconstructed waveforms against the injected ones, then averaging the results across different detectors. The resulting H_0 is not exactly the null hypothesis that describes the on-source distance measurement: in fact, in on-source the comparison is between the burst reconstruction and the best CBC model, the MAP, since the actual signal is of course unknown. This asymmetry is one of the plausible causes for the relative abundance of larger p-values in the waveform consistency tests in LVK catalogs [41]. This problem is not addressed by this work.

The overlap as metric of waveform distance suffers from other problems. It neglects all types of correlation in the waveform of the signal by simply summing over the time samples. At the same time, it measures just the angle between two waveforms, being completely blind to errors affecting their amplitude scales. Lastly, it is defined $\in [-1, 1]$, so that it may suffer from saturation effects of its values.

The NCM, instead, is designed to account for the correlation structure of the waveforms, both per time samples in the single detector and in the network. The Mahalanobis distance values are not limited to a specific range, $\mathcal{M}^2 \in \mathbb{R}^+$, and properly weights the contributions from different detectors according to the NCM.

Our objective here is to define a waveform consistency test based on $\mathcal{M}^2$. The procedure is built by "reversing" the procedure with respect to what has been done in the previous two sections.
The first step consists in injecting the waveform representative of the null hypothesis H_0 (here the MAP estimate) in the off-source data several times and use pycWB to provide the table of reconstructed waveforms for the network. To estimate the Mahalanobis distance, the full set is split in two disjoint subsets. 50% of injections are used to compute the average of the reconstructions $\bar{h}_{H_0}$, and the NCM C. The other 50% is used to build the distribution of the $\mathcal{M}^2$ under H_0 as:

$$\mathcal{M}^2 = (h^*_{H_0} - \bar{h}_{H_0})^T C^+ (h^*_{H_0} - \bar{h}_{H_0}) \tag{8.18}$$

with $h^*_{H_0}$ the network reconstruction of the injected GR template[6]. The p-value is then computed by comparing such $\mathcal{M}^2$ distribution with the observed value on the on-source reconstruction $\hat{h}$ as:

$$\mathcal{M}^2_{obs} = (\hat{h}^- \bar{h}_{H_0})^T C^+ (\hat{h}^- \bar{h}_{H_0}) \tag{8.19}$$

Notice that, if the residuals $h^*_{H_0} - \bar{h}_{H_0}$ were gaussian, the null hypothesis distribution would be a χ^2 distribution with N_{df} degrees of freedom as determined via the principal component analysis. Being the χ^2 distribution the uniformly most powerful distribution for the null hypothesis, the test based on $\mathcal{M}^2$ would be the uniformly most powerful test for this problem.

Unfortunately, the residuals $h^*_{obs} - h_{H_0}$ are not gaussian, so that the real distribution of the test statistics under the null hypothesis can deviate from a χ^2 distribution, with significant differences on the estimate of the p-value for the reconstruction. Both p-values are be reported in this section as an informal measure of the difference between the two distributions and of the importance of accounting for the real distribution of the test statistic in our specific case study under the null hypothesis.

For the case of GW250114, we perform this test by injecting the GR template obtained from the MAP estimate in real noise realizations and reconstructing it with pycWB. For a complete comparison, we perform the test and draw the p-value from both the χ^2 distribution and the empirical distribution obtained by the off-source signal injections. We also compare to the LVK result obtained in GWTC5 [43] using the overlap as a test statistic, comparing the distribution of the overlap obtained by the injections with the observed value of the overlap for the on-source reconstruction. The results of this analysis are summarized in Figure 8.10. The p-values for the three different cases are shown in Table 8.2.

It is interesting to compare the results obtained with the three test statistics. Both Figure 8.10 and Table 8.2 show that the the χ^2 distribution is not a suitable approximation for the real distribution of the test statistic under the null hypothesis. In fact, the real statistics appear to be much larger than the one expected from the χ^2 distribution, especially in the right tail. This results on two very different declared p-values, with a p-value of $p \simeq 0$ for the

[6]Notice the difference with the previous cases. In sections 8.3.1 and 8.3.2, the NCM and the Mahalanobis reference were built on top of the bootstrap samples, and the CBC models where compared against these references. In this case, we build a NCM and the reference from the CBC samples and use to on-source estimate to compute where the observation sits of the distribution built by CBC samples

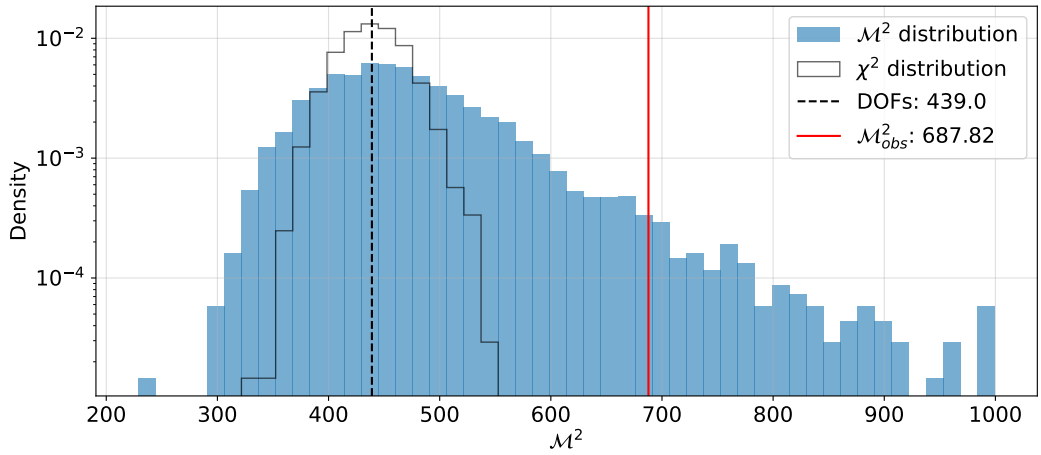


Figure 8.10: Distribution of the test statistic $\mathcal{M}^2$ under the null hypothesis H_0. The Blue histogram shows the empirical distribution, while the black histogram shows the χ^2 approximation. The red, vertical line represents the observed value of the test statistic for the on-source reconstruction of GW250114.

p-value		
χ^2	$\mathcal{M}^2$	overlap
0	0.031	0.664

Table 8.2: P-values for the waveform consistency test on GW250114. The p-values are computed using the empirical distribution for $\mathcal{M}^2$, the overlap and the χ^2 approximation.

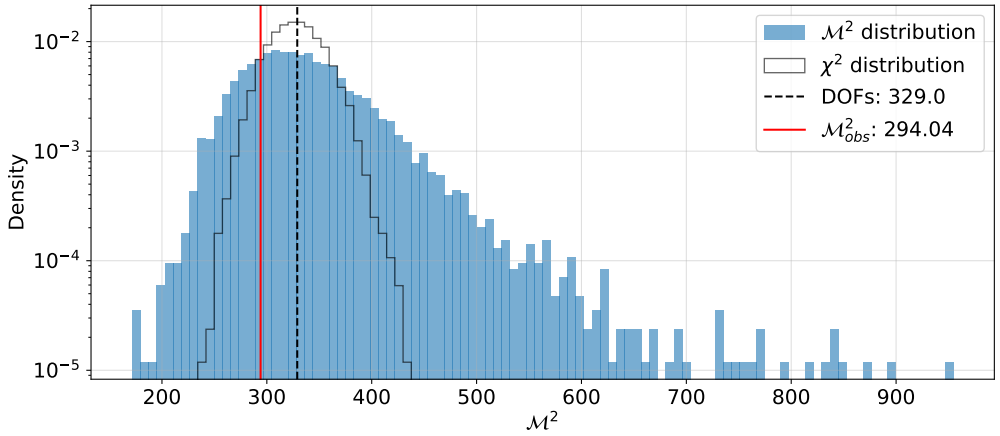


Figure 8.11: Distribution of the null statistics $\mathcal{M}^2$ under the hypothesis of the data being described by the whole set of the CBC samples. The Blue histogram shows the empirical distribution, while the black histogram shows the χ^2 approximation. The red, vertical line represents the observed value of the test statistic for the on-source reconstruction of GW250114.

χ^2 approximation against the value of ~ 0.03 obtained when the empirical distribution is taken into account. Therefore, for practical applications, the empirical calibration remains the only choice for p-value estimation and the χ^2 approximation is far to be met.
For the overlap, the p-value obtained with this test statistic is much larger than the one obtained with the NCM, with a value of $p \simeq .664$. In the standard analysis, the overlap is computed by injecting waveforms that are not only representative of the MAP, but the posterior samples as a whole. For this reason, we have also investigated the result provided by $\mathcal{M}^2$ in this scenario. The results are shown in Figure 8.11 and the estimated p-value is 0.77, compatible with the expectations.
More tests are needed to assess if the $\mathcal{M}^2$ test is compliant with the expectation of p-values uniformly random distributed under the null hypothesis.

8.5 Conclusions

In this last, conclusive chapter we have shown how the bootstrap reconstructions can be used to provide way more information that the one contained in the mere reconstruction of the error around the waveform estimate, and how this information can be encoded in a clean, compact and powerful way in the form of a "Network Covariance Matrix" (NCM).
The NCM is a new tool that we have developed in context of minimally modeled analyses and encodes all the information about the main error sources in our reconstruction. One of the big advantages of building a Network Covariance Matrix, is that it provides a natural way of combining all the detectors in the network, and can be generalized to any number of detectors. This method automatically solves two major sources of problems that have not been properly addressed in the past and in the existing literature. Firstly it shows how to correctly deal with the correlation structure of the detectors into the network and how to correctly combine the information from the different detectors. Secondly, it completely takes into account the correlation structure of the reconstruction inside each and every detector

independently. This enables determining the real number of degrees of freedom that we have in our reconstruction, which is significantly smaller than the number of time samples. Under the assumption of Gaussian noise and implementing a bootstrap technique to provide independent instances of the local noise, the total number of the degrees of freedom is roughly $\sim 1/5$ the number of time samples at 2kHz in the selected science case of the high SNR event GW250114.

Even more interestingly, once the relevant degrees of freedom have been selected with some principle (we used a threshold of 99% of the explained variance), the NCM can be inverted and used to define a distance between waveforms. We adopted the Mahalanobis distance, a widely used definiton of distance which usually has good properties. The NCM is a building block that weights the distance between two "network waveforms", taking completely into account the correlation structure of the problem. We have shown how this distance can be used to build statistical inference in three different contexts: parameter estimation, model selection and hypothesis testing. In the first case, we have been able to mitigate the bimodality of the posterior marginal probabilities from the CBC parameter estimation on the sky localization parameters. However, in case of CBC signals, the $\mathcal{M}^2$ method will not be able to compete in general with modeled CBC parameter estimation methods, and we explored this application only for testing purposes.

Much more interestingly, we have shown in a "purely-academic" example how the NCM can be used to perform model selection when we have two alternative, competing hypothesis. We considered a case in which the null hypothesis is a pure GR model, while the alternative hypothesis is a non-GR model obtained by including one non-GR post newtonian coefficient. We compared this results with the standard method used in the literature to compute the "distance" between waveforms and perform model selection: the overlap. The improvement in the power of the test is significant in a relevant range of false alarm probabilities. Despite this exercise is not expected to be representative of what we would observe in a real deviation from GR, we think we provided enough evidence of the potential of the NCM framework. Testing in a variety of different conditions will be anyway necessary to draw any general, quantitative conclusion. Generally speaking, this framework could actually be useful to discriminate between different types of hypothesis whenever there are no strong priors, for example in the observation of a CCSNe.

The NCM framework can be used to perform a more standard hypothesis testing, such as in the case of waveform consistency tests, which are already reported in the LVK collaboration papers. We have shown how to build the null-hypothesis distribution for similar tests, by injecting the GR template of the null hypothesis in different noise realizations and computing the network covariance matrix for this reconstruction. We have shown how to compute the p-value for the observed value of the test statistic, and we have compared it with the p-value obtained with the standard method based on the overlap. However, the result from one single experiment is not sufficient to demonstrate the correctness of this test, i.e. the random uniform distribution of p-value estimates under the null hypothesis. To draw more precise conclusion, one should analyze several events with both methods and check the pp-plot for the two methods.
Nevertheless, we are convinced we provided enough evidence of the high potential of this framework, that could address numerous questions and problems that were not properly addressed before in the context of burst searches, and could be a powerful tool for future applications.

Conclusions

Burst methods for gravitational-wave (GW) detection and reconstruction play a central role in GW data analysis. Their main strength is that they do not rely on a specific theoretical waveform model for either signal detection or reconstruction. Although some assumptions are still required, these methods remain applicable to a broad class of signals with minimal prior information. The central focus of this thesis has been the reconstruction of gravitational wave waveforms in the context of unmodeled burst searches with the pycWB pipeline. The waveform reconstruction represents a crucial component of gravitational wave data analysis, as it carries the physical interpretation of the GW emission mechanism. This aspect is particularly relevant for burst searches, which can target an arbitrary class of sources and are not restricted to specific waveform models. Due to the broadness of this scope, the present investigation has been organized into three complementary and consequential parts: the assessment of reconstruction accuracy, the characterization of reconstruction uncertainties, and the development of a framework for statistical inference using the estimated waveforms. The first part of the thesis concentrated on improving the accuracy of waveform reconstruction by investigating two distinct and complementary modifications to the pycWB pipeline. The first modification targeted the wavelet tuning of the WDM-wavelet transform, choosing a tuning that improves the resolution in the time domain with a loss of resolution in the frequency domain. This modification results in a more balanced trade-off between time and frequency resolution, enabling a more compact representation in the time-frequency plane, bringing the wavelet basis closer to the Heisenberg-Gabor limit by approximately 20%. The second modification introduced a data whitening procedure based on the Maximum Entropy Spectral Analysis (MESA) method. With this spectral estimator, we have built an alternative whitening procedure to the pycWB standard. Together with the wavelet tuning, we systematically assessed the combined impact of both modifications on the quality of waveform reconstruction by injecting signals across a broad range of sky positions and different signal morphologies. Although MESA showed several advantages in preliminary analysis, such as a more representative estimate of the PSDs, improved whitening, and less time leakage, these improvements are sub-dominant compared to other artifacts introduced by the pipeline. In contrast, the results of this analysis reveal that the impact of the WDM tuning is small but evident and produces measurable improvements in the accuracy of the reconstruction process. Throughout this analysis, particular attention was devoted to time-leakage effects, which represent the contamination of high signal-to-noise ratio (SNR) components from one time interval into neighboring regions. As demonstrated in Chapter 5, the phenomenon of leakage arises from two distinct sources: first, the non-diagonality of whitening filters in the time domain; second, the non-local nature of the WDM basis functions. These two sources of leakage interact in complex ways, with the effects related to the WDM dominating over those arising from the whitening filter. Suitable methods have been added to the pycWB pipeline to exploit these results.

The second part of this work focused on the characterization and quantification of uncertainties in reconstructed waveforms. To achieve this, we implemented a Bootstrap-based approach. Bootstrap techniques represent a novel development within burst analyses, as they estimate reconstruction uncertainties through the statistical characterization of the residual-noise distribution. We implemented and compared two approaches based on the bootstrap method: the parametric bootstrap and the non-parametric bootstrap. Both methods were designed to construct confidence intervals around the pycWB point estimate of the waveform. The parametric bootstrap relies on the assumption that the residuals after signal removal follow a multivariate normal distribution, and generates bootstrap replicates by injecting the on-source reconstruction into synthetic stationary Gaussian noise with a power spectrum estimated from the data. The non-parametric bootstrap operates directly on the whitened residuals and generates bootstrap samples through direct resampling with replacement. Our investigation compared these two bootstrap approaches against a reference off-source experiment, in which the same signal is re-injected multiple times into stationary gaussian noise. The results show that both bootstrap methods provide comparable estimates of the reference confidence intervals. Importantly, within the scope of our investigation, neither approach showed any clear preferential behavior with respect to the off-source experiment. While the non-parametric bootstrap embodies the principle of model-independent inference desired in burst searches, the parametric bootstrap integrates more naturally into the existing pycWB workflow. For these practical reasons, the parametric bootstrap was selected for implementation in the analysis pipeline and used throughout the subsequent investigations presented in this thesis.

The third and final part of this thesis introduced a novel framework for statistical inference based on the Network Covariance Matrix (NCM). Beyond its use for the construction of confidence intervals as discussed in the previous chapter, NCM is a more general method for extracting statistical information from the ensemble of waveform reconstructions obtained via bootstrap analysis. The Network Covariance Matrix is constructed by computing the sample covariance of the parametric bootstrap replicates across all time samples and all detectors in the network. This matrix encodes the complete structure of correlations both within the reconstruction from each individual detector and across the detectors comprising the network. A fundamental characteristic of this matrix is that the effective degrees of freedom in the network reconstruction are typically smaller than the total number of time samples, reflecting the intrinsic correlations introduced by the coherent analysis procedure. To extract and quantify these degrees of freedom, we apply Principal Component Analysis to diagonalize the NCM, obtaining the eigenvalue decomposition that reveals the true dimensionality of the problem. This decomposition provides an orthonormal basis of eigenmodes whose eigenvalues characterize how the statistical uncertainty is distributed across the various degrees of freedom. More interestingly, some of the eigenmodes can shed light on inner mechanisms of the burst reconstruction procedure.

To explore the practical utility and power of the NCM framework, we tested it on a science case, selecting the GW event GW250114, a highly symmetric binary black hole merger with a total mass of approximately 70 solar masses and total network signal-to-noise ratio of 73.5 for Burst Analysis. This event provided an ideal test case due to its high signal-to-noise ratio and well-characterized properties. Through systematic application of the NCM framework and construction of the associated bootstrap replicates, we demonstrated three distinct applications of the method. All the three methods are based on the computation of the Mahalanobis distance, a multivariate distance metric that uses the NCM to weight the distance between the burst reconstruction and the model under investigation.

The first application explored the use of the NCM framework for parameter estimation.

Our investigation is purely exploratory. We examined the sensitivity of our reconstruction to variations in signal parameters by computing the Mahalanobis distance between the on-source reconstruction and modeled waveforms generated with different parameter samples from the CBC posterior distribution. Notably, while the full CBC posterior distribution exhibits a pronounced bimodality in the marginal distribution of the sky location parameters, the subset of parameters that minimize the Mahalanobis distance shows a strong mitigation of this bimodality by selecting the mode in which the MAP estimate actually lies. For this specific case, the Mahalanobis distance is very sensitive to the sky location parameters, since it is strongly correlated (almost 70%) to these parameters, contrary to the log-posterior from CBC parameter estimation. This difference between the two families of estimators is also reflected by the very low correlation between Mahalanobis distance and log-posterior probability. We cannot conclude here that NCM framework successfully helps disambiguate multimodality in sky error regions, having tested only a single, exceptional event with an high SNR. While we do not expect our method to be competitive with standard CBC parameter estimation in general, these initial hints suggest it can be useful in specific scenarios.

The second application was an exercise in model selection that involved building the distributions of two competing hypotheses against the on-source reconstruction. The two competing models were the General Relativity (GR) model, in which the null hypothesis is built by varying the model parameters within the CBC posterior distribution of GW250114, and an unphysical alternative hypothesis represented by a 0 order post Newtonian modification of the inspiral based on a specific parameter (dchi0). The two distributions were built by computing the Mahalanobis distance between the on-source reconstruction and the modeled waveforms generated with different parameter samples from the CBC posterior distribution for the GR model, and with different values of the dchi0 parameter for the alternative model. We compared these distributions for the NCM-based Mahalanobis distance test statistic against two standard approaches used in LVK analyses: the overlap statistic, which measures the inner product between reconstructed and modeled waveforms and serves as the current standard for waveform consistency through burst methods, and the naive χ^2 test, which sums squared residuals across all time samples while treating each sample as independent. For the lower values of the Type I error rate, the NCM method exhibits power improvements ranging from approximately 5 to 20 times in the type I range below $\sim 1\%$.

The third application addressed waveform consistency testing, a fundamental procedure in burst searches designed to assess whether a reconstructed waveform is consistent with the predictions of signal models. In our test, we use the burst reconstruction to estimate a p-value for the null hypothesis of the signal being correctly described by the GR model estimated by the Maximum a Posteriori. This is a preliminary test to explore the performances of $\mathcal{M}^2$ statistics in different inference contexts.

In summary, through Chapters 5 to 7, this thesis has developed and systematically investigated multiple innovations aimed at improving both the accuracy and the statistical treatment of gravitational waveform reconstruction in burst searches. The reduction of the WDM β order yields measurable improvements in reconstruction accuracy by enhancing time-frequency resolution. The implementation of bootstrap methods provides principled uncertainty quantification without strong distributional assumptions. In the last chapter, we started exploring the concept of the "Network Covariance Matrix", a framework that provides a general, extensible methodology for statistical inference that properly accounts for the true correlation structure of coherent network reconstructions. These developments have been validated through application to public LVK collaboration gravitational wave data and demonstrate clear advantages over existing approaches in specific scenarios. However, as detailed in Section 8.4 of the case study chapter, more extensive investigation is needed to

fully explore the potential and limitations of these methods across diverse scenarios. Future work should prioritize a systematic validation of the NCM methodology across a broader set of astrophysical sources, along with a detailed investigation of how an expanded detector network, including Virgo and other future observatories, can improve both the statistical inference and tests of fundamental physics.

Appendix A

Levinson Recursion

A.1 Levinson Recursion

The solution of MESA for the Power Spectral Density is a closed form solution:

$$S(f) = \frac{P_N \Delta_t}{\left(\sum_{s=0}^{N} a_s z^z\right)\left(\sum_{s=0}^{N} a_s^* z^{-s}\right)}. \tag{A.1}$$

in terms of the prediction error filter coefficients a_i (with $a_0 = 1$) and the prediction error P_N. We now want to show how these coefficients are computed. To compute them a_i-s, we substitute in equation (4.5) with the above representation for $S(f)$ in terms of the a_i coefficients. Changing variable from f to $z(f) = e^{-\imath 2\pi f \Delta t}$, the integral (4.5) becomes a contour integral in the unit circle centered at the origin of the complex plane:

$$\frac{P_N}{2\pi\imath} \oint_{\mathbb{S}^1} \frac{z^{-s-1}}{\sum_{n=0}^{N} a_n z^n \sum_{n=0}^{N} a_n^* z^{-n}} = \bar{r}_s. \tag{A.2}$$

Sending $s \to s-r$, multiplying by a_s^* and summing over all possible values for s the equation becomes

$$\begin{aligned} \sum_{s=0}^{N} a_s \bar{r}_{s-r} &= \frac{P_N}{2\pi\imath} \oint \frac{z^{r-1} \sum_{s=0}^{N} a^* s z^{-s}}{\sum_{s=0}^{N} a_s z^s \sum_{s=0}^{N} a^* s z^{-s}} dz \\ &= \frac{P_N}{2\pi\imath} \oint \frac{z^{r-1}}{\sum_{s=0}^{N} a_s z^s} dz \end{aligned} \tag{A.3}$$

Since all the zeros of the denominator lay outside the unit circle, the contour integral is 0. The only exception is for $r = 0$ with a residual in the origin of the axis. For $r = 0$ the integral can be computed via the Cauchy integral formula and (A.3) becomes:

$$\sum_{s=0}^{N} a_s \bar{r}_{r-s} = P_N \qquad \text{if } r = 0 \tag{A.4}$$

$$\sum_{s=0}^{N} a_s \bar{r}_{r-s} = 0 \qquad \text{if } r \neq 0. \tag{A.5}$$

The equations (A.4) and (A.5) can be rewritten as a Matrix equation of the form

$$\begin{pmatrix} \bar{r}_0 & \bar{r}_{-1} & \dots & \bar{r}_{-N} \\ \bar{r}_1 & \bar{r}_0 & \dots & \bar{r}_{1-N} \\ \bar{r}_2 & \bar{r}_1 & \dots & \bar{r}_{2-N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{r}_N & \bar{r}_{N-1} & \dots & \bar{r}_0 \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} P_N \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{A.6}$$

know as the prediction error filter equation. It is a matrix equation involing the Toeplitz Matrix R_{ij} and the vector $\vec{a}$, known as the forward prediction error filter. The approach for solving the equations is the Levinson - Durbin recursion.
Our goal is to obtain the N-th order equation (A.6) from the $N-1$-th order equation:

$$\begin{pmatrix} \bar{r}_0 & \bar{r}_{-1} & \dots & \bar{r}_{1-N} \\ \bar{r}_1 & \bar{r}_0 & \dots & \bar{r}_{2-N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{r}_{N-1} & \bar{r}_{N-2} & \dots & \bar{r}_0 \end{pmatrix} \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_{N-1} \end{pmatrix} = \begin{pmatrix} P_{N-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{A.7}$$

This equation is not sufficient to solve the recursion. Since the autocorrelation matrix is self-adjoint, from equations (A.4) and (A.5) and noting that the first and the last row of the autocorrelation matrix are respectively the complex conjugate reverse, is easy to show that

$$\begin{pmatrix} \bar{r}_0 & \bar{r}_{-1} & \dots & \bar{r}_{1-N} \\ \bar{r}_1 & \bar{r}_0 & \dots & \bar{r}_{2-N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{r}_{N-1} & \bar{r}_{N-2} & \dots & \bar{r}_0 \end{pmatrix} \begin{pmatrix} b^*_{N-1} \\ b^*_{N-2} \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ P_{N-1} \end{pmatrix} \tag{A.8}$$

holds. The vector obtained reflecting b and taking its complex conjugate is known as "backward prediction error".
Stepping from order $N-1$ to order N, the autocorrelation matrix grows from size (N, N) to size $(N+1, N+1)$, this implies that a new component b_n for the prediction error filter is needed. If we take it to be $b_N = 0$, this choice does not modify the first N component of the right side of equation (A.7) (or the last N components of (A.8)), i.e.:

$$\begin{pmatrix} \bar{r}_0 & \bar{r}_{-1} & \dots & \bar{r}_{1-N} & \bar{r}_{-N} \\ \bar{r}_1 & \bar{r}_0 & \dots & \bar{r}_{2-N} & \bar{r}_{1-N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{r}_{N-1} & \bar{r}_{N-2} & \dots & \bar{r}_0 & \bar{r}_{-1} \\ \bar{r}_N & \bar{r}_{N-1} & \dots & \bar{r}_1 & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_{N-1} \\ 0 \end{pmatrix} = \begin{pmatrix} P_{N-1} \\ 0 \\ \vdots \\ 0 \\ \Delta_N \end{pmatrix}. \tag{A.9}$$

with

$$\Delta_N = \sum_{n=0}^{N} \bar{r}_{N-n} b_n. \tag{A.10}$$

Adding the previous expansion for both forward and backward equations, one obtains

$$\begin{pmatrix} \bar{r}_0 & \bar{r}_{-1} & \dots & \bar{r}_{-N} \\ \bar{r}_1 & \bar{r}_0 & \dots & \bar{r}_{1-N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{r}_{N-1} & \bar{r}_{N-2} & \dots & \bar{r}_{-1} \\ \bar{r}_N & \bar{r}_{N-1} & \dots & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_{N-1} \\ 0 \end{pmatrix} + c_N \begin{pmatrix} 0 \\ b^*_{N-1} \\ \vdots \\ b_1 \\ 1 \end{pmatrix} = \begin{pmatrix} P_{N-1} \\ 0 \\ \vdots \\ 0 \\ \Delta_N \end{pmatrix} . + c_N \begin{pmatrix} \Delta^*_N \\ 0 \\ \vdots \\ 0 \\ P_{N-1} \end{pmatrix}.$$

With the reflection coefficient c_N to be computed. Requiring the previous equation to be equivalent to A.6, one obtains

$$\begin{cases} P_{N-1} + c_N \Delta_N^* = P_N \\ c_N P_{N-1} + \Delta_N = 0 \end{cases}$$

which can be easily solved for c_N and P_N as

$$c_N = -\frac{\Delta_N}{P_{N-1}}; \quad \text{and } P_N = P_{N-1}\left(1 - |c_N|^2\right). \tag{A.11}$$

Having computed the value for the reflection coefficient, the N-th order forward prediction error filter is found to be

$$a_i = b_i + c_N b_{N-i}^*. \tag{A.12}$$

To obtain the solution for the N-th order, one starts $\bar{r}_0 = P_0$ and recursively computes the new values until the desired order is computed. This problem is often solved numerically with Burg's Algorithm [119].

Appendix B

MESA Whitening Algorithm

In thix appendix we report the scheme of the algorithm that implements whitening based on MESA. It is available in the pycWB repository [123].

Step 1: Pre-processing

1.1 The input strain data is optionally converted to a PyCBC TimeSeries object for consistent handling.

1.2 The mean of the time series is subtracted to center the signal around zero.

1.3 A high-pass Butterworth filter is applied to remove low-frequency noise, with a cutoff defined by the configuration (typically f_{cut} = 16 Hz).

Step 2: Windowed Whitening with MESA

2.1 The data is divided into overlapping 15-second windows, with a stride of 5 seconds between each window. This allows for time-local whitening, mitigating non-stationary behavior.

2.2 For each window, the Maximum Entropy Spectral Analysis (MESA) is used to estimate the Power Spectral Density (PSD). The autoregressive order and solver type are specified in the configuration.

2.3 The PSD is smoothed taking the median over 9 PSD estimates, centered in the central "stride" seconds

2.3 Each window is whitened in the frequency domain by dividing the Fourier-transformed data by the square root of the estimated PSD. It is then transformed back to the time domain.

2.4 Only the central central "stride" seconds of each whitened window are retained. These pieces are stitched together across the full time series, ensuring that the output maintains temporal continuity without edge artifacts.

Step 3: Outlier Detection and PSD Reindexing (Optional)

3.1 If enabled, an Isolation Forest algorithm is used to identify anomalous PSD estimates across windows.

3.2 Features used for anomaly detection include the deviation of each PSD from the median in both low-frequency (16--128 Hz) and high-frequency (above 128 Hz) ranges.

3.3 PSDs classified as outliers are replaced with their nearest in-time, non-anomalous neighbor to reduce the impact of glitches on whitening quality.

Step 4: nRMS Computation (Reverse Engineering the Whitening)

4.1 Time-frequency maps of both the original and whitened data are constructed using a wavelet transform.

4.2 The ratio of the unwhitened to whitened map defines the whitening filter's effect in time-frequency space.

4.3 This whitening effect (the nRMS matrix) is reshaped into 20-second segments per frequency bin.

4.4 For each frequency bin and segment, the root-mean-square is computed across time.

4.5 Amplitudes below f_{cut}(16 Hz) are manually set to 1 for compatibility.

4.6 The resulting nRMS vector encodes the amplitude normalization applied to the signal and is required in future stages of the analysis.

Appendix C

Isolation Forest

C.1 Isolation Forest

In machine learning, several techniques exist to detect outliers in a dataset. One noteworthy algorithm is the so-called **Isolation Forest** [80]. The implementation used here is the one provided by the `SKLearn` Python package [54].

Let us define our dataset as X_{ij}, where $i = 1, \ldots, n$ denotes the data samples and $j = 1, \ldots, p$ the features of each sample.

An Isolation Forest identifies outliers by constructing multiple **isolation trees**. The algorithm is based on two key assumptions:

- Anomalies are a minority ($< 50\%$);
- Their feature values significantly differ from those of normal data.

Both assumptions align with the common-sense notion of an outlier, though only the first offers a quantitative criterion. In our case, the first assumption is expected to hold, as the dataset has undergone Data Quality checks, and only "Good Quality Data" are retained. The second assumption is more qualitative, requiring an appropriate choice of features. To this end, several definitions for the features (r_{lf}, r_{hf}) were explored, and those presented in Equation [5.24] were selected because they maximize the separation between outliers and inliers.

Compared to other outlier detection algorithms (e.g., Elliptical Envelopes, Local Outlier Factor), the Isolation Forest offers several advantages: it is well-suited to small datasets (in our case, around 60 samples) and has an algorithmic complexity of $\mathcal{O}(n)$, scaling linearly with the number of data samples.

The Isolation Forest consists of a collection of isolation trees, each constructed as follows: The root node initially contains a number n of samples. Then, recursively:

1. A random feature f_i is selected;
2. A random threshold value is chosen for that feature from a uniform distribution between its minimum and maximum values within the current node;
3. Two child nodes are created, containing the data with feature values below and above the threshold, respectively.

This process is repeated at each level of the tree until one of the following conditions is met:

- A leaf node contains only one data point, in which case it is no longer split;
- A predefined maximum depth D is reached, and the recursion stops.

The maximum depth is defined as $D = \lceil \log_2(n) \rceil$, which in our case yields $D \sim 6$.

Once an isolation tree is constructed, for any given data point x, its path length $h(x)$ is defined as the number of edges traversed from the root node to the leaf node where x falls.

The Isolation Forest contains $N_{\text{trees}} = 100$ instances of trees which are combined to provide an evaluation score s for every data point x, defined as:

$$s(x, n) = 2^{-\frac{\mathbb{E}[h(x)]}{c(n)}} \tag{C.1}$$

where $\mathbb{E}[h(x)]$ is the average path length of the data point x computed over all the isolation trees in the forest, and $c(n)$ is a normalization factor defined as the average path length of an unsuccessful search in a Binary Search Tree:

$$c(n) = 2 \sum_{k=1}^{n-1} \frac{1}{k} - 2 \left(\frac{n-1}{n} \right). \tag{C.2}$$

The following cases hold:

- if x is not an anomaly, it requires more splits to be isolated, so $\mathbb{E}[h(x)] > c(n)$ and $s \to 0$;
- if x is an anomaly, it is isolated quickly, so $\mathbb{E}[h(x)] < c(n)$ and $s \to 1$.

The skLearn packages makes additional adjustments to the score, let's call it s^*, such that:

$$\begin{cases} s^* > 0 \text{ for inliers} \\ s^* < 0 \text{ for outliers} \end{cases}. \tag{C.3}$$

Figure C.1 reports the result of the classification of the Isolation forest algorithm for the example in Section 5.4.2.

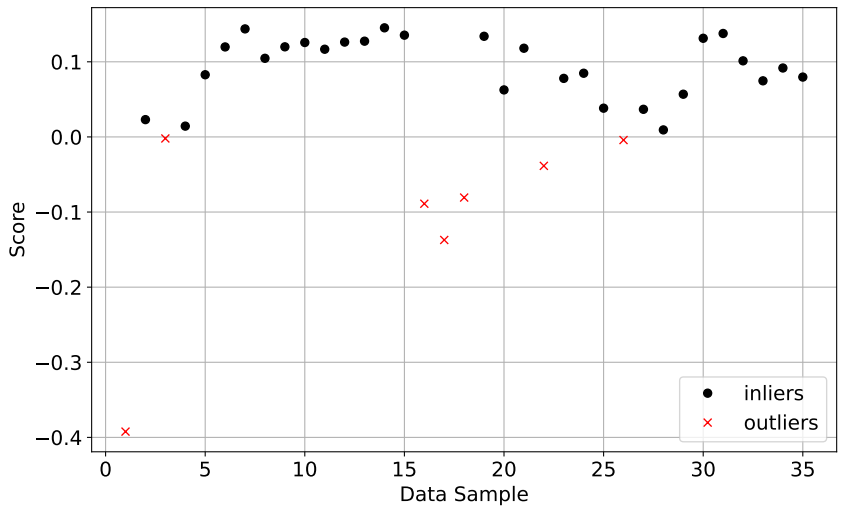


Figure C.1: Values, as a function of the data sample, of the Isolation Forest score as implemented in scikit-learn.

Appendix D

Outliers in the whitening tests

In this appendix we report a short comparison of the outliers of the distributions studied in chapter 5.5.1. We defined the outliers in the section as either p-values that are exactly equal to zero (which represent approximately 3% of the data), or points that fall far (outside 5σ) from the bulk of the distribution for the mean and standard deaviation of the whitened time series. These outliers are sparse and are mostly compatible with data segments contaminated by loud glitches. In this appendix we show the Q-transform of some of the relevant outliers, to highlight their nature. We display some of them using a Q-transform obtained with gwpy [83]. The results of the Q-transform are shown in Figure D.1. The Q-Transform is goes from $-0.5s$ to $0.5s$ around the time of the outlier, defined as the time at which the maximum value for the data is found. In all the plots the energy is extremely large, and the background noise energy is negligible with respect to the energy of the outlier, whose energy goes up to 10^4 times the background noise. For a comparison over a larger number of outliers, we compare the distribution of the maximum amplitude of the whitened data that are considered outliers with the distribution of the maximum amplitude of the whitened data that are not considered outliers. The results are shown in Figure D.2. As is is evident from the figures, the segments discarded as otuliers are segments that contain extreme values for the maximum amplitude of the whitened data when compared to the segments that are not considered outliers. Therefore, most of these segments appear compatible with a contamination of the data by loud glitches.

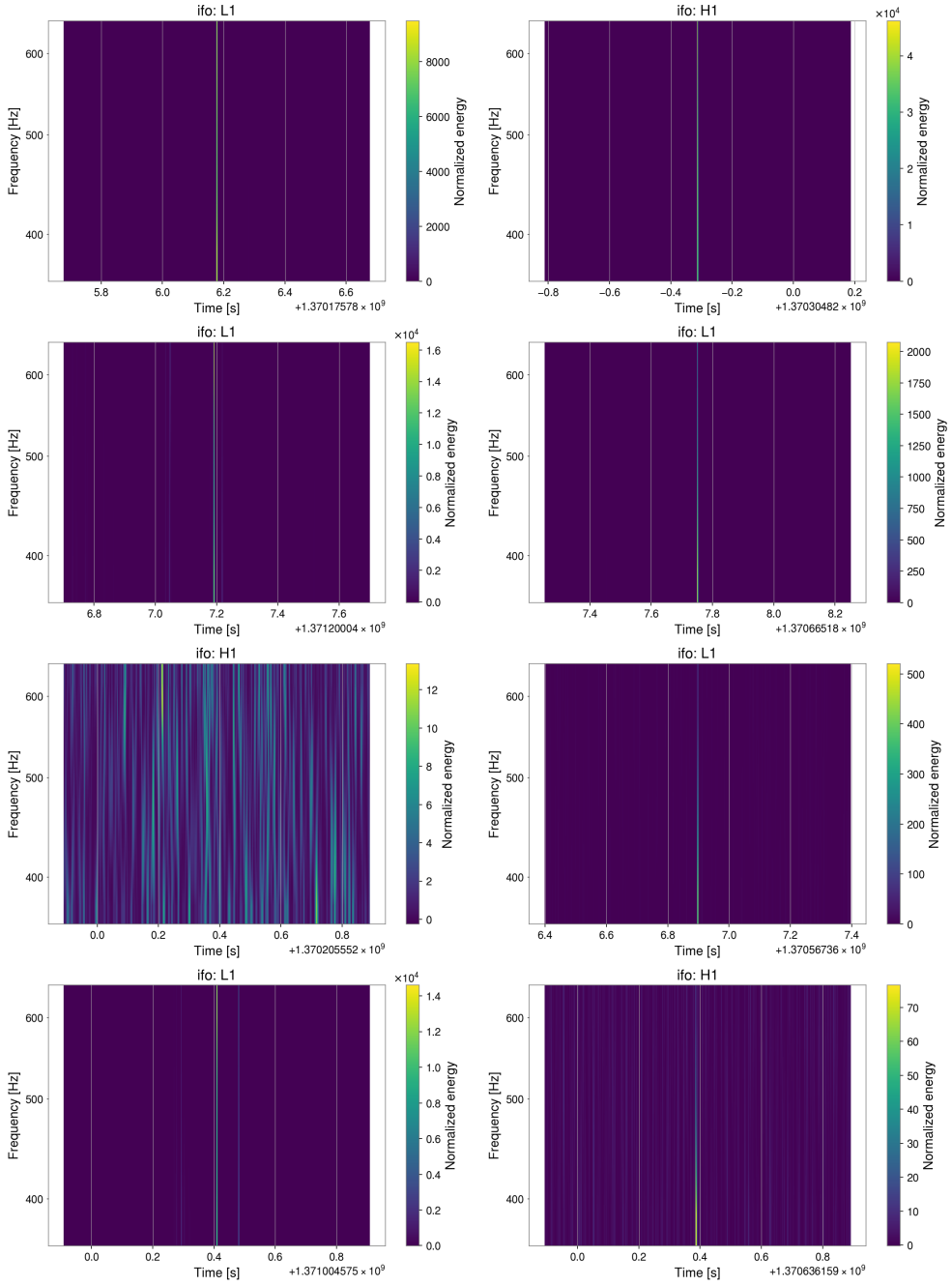


Figure D.1: Q-transform of some of the outliers found in the p-values distribution. The Q-transform is computed using gwpy [83]. The transform ranges the 1 second arount the time at which the maximum oscillation of the whitened data occurs

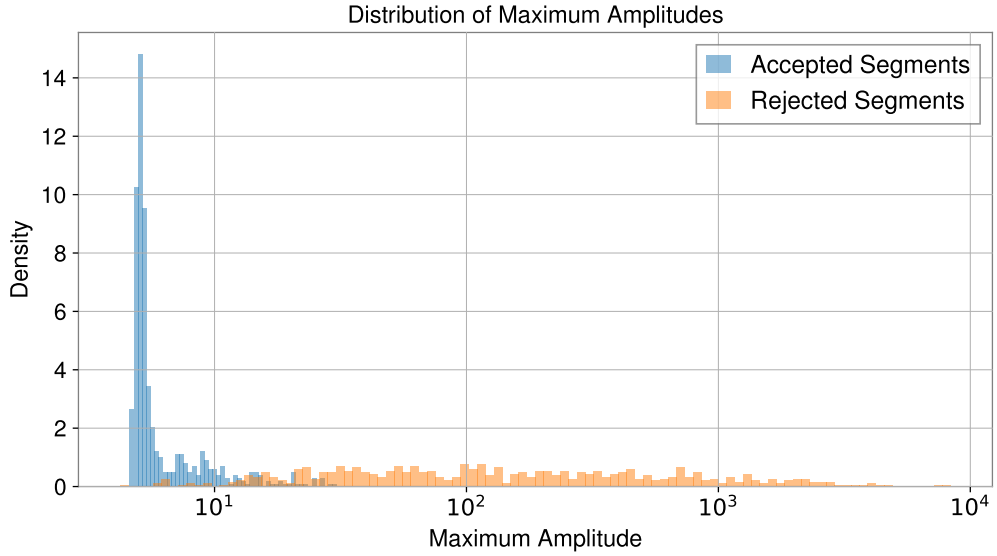


Figure D.2: Distribution of the maximum amplitude of the whitened data for outliers and non-outliers.

Appendix E

Case example of waveform reconstruction for CBC signals

In this appendix we report two examples comparing waveform reconstruction obtained with MESA and standard pycWB whitening. In both cases we consider the injection and reconstruction of CBC signals generated with the `pycbc` library using the `IMRPhenomD` waveform model [98], which provides a complete description of the inspiral, merger, and ringdown phases of binary black hole coalescences.

For each configuration we perform approximately 3000 injections on real detector data from a fixed sky location, in order to build an ensemble of waveform reconstructions and statistically compare the two whitening methods. The injections are placed at latitude $+50°$ and longitude $-110°$, close to the maximum of the antenna pattern for the HL network, so as to maximize the detection efficiency and achieve a high SNR. The analysis compares the standard pycWB reconstruction with $\beta = 6$ against the MESA reconstruction with $ar = 500$ and $\beta = 2$.

To compare the reconstructed waveforms with the injected signals, we focus on two quantities. The first is the overlap between the reconstructed waveform and the injected signal. The second is the cumulative hrss of the reconstructed waveform over time, normalized by the total hrss of the injected waveform, and compared against the injected cumulative hrss. We choose this representation instead of directly comparing the waveforms for several reasons. First, being monotonic, the cumulative hrss is easier to visualize and interpret. Second, it allows both local and global differences in the reconstructed signal energy to be quantified, providing a more complete characterization of the reconstruction performance.

The two injected signals are chosen to represent two extreme regions of the CBC parameter space. The first is a low-mass binary with component masses $m_1 = 20M_\odot$ and $m_2 = 15M_\odot$ at a distance of 2 Gpc. The second is a high-mass binary with component masses $m_1 = 110M_\odot$ and $m_2 = 90M_\odot$ at a distance of 2.5 Gpc. The low-mass system has a longer duration in the detector band and exhibits several inspiral cycles, while the high-mass system is shorter and dominated by the merger and ringdown phases. These two cases therefore allow us to separately investigate the reconstruction of weak inspiral features and of short, high-energy merger/ringdown signals.

The inspiral regime is particularly interesting because its local SNR is significantly lower than that of the merger and ringdown phases. The capability to accurately reconstruct a long inspiral is therefore representative of the pipeline sensitivity to small waveform features, which is especially relevant when searching for weak physical effects or deviations

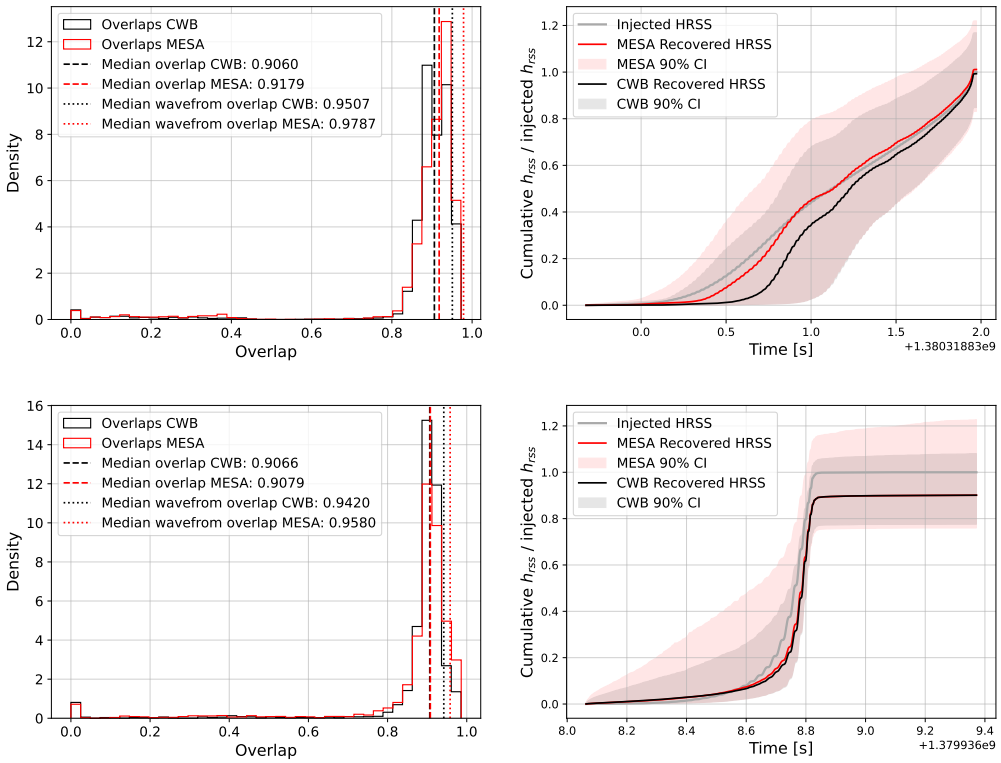


Figure E.1: Overlap of the reconstructed waveform with the injected signal for the high-mass binary (left) and the reconstructed cumulative hrss over time (right), normalized by the total injected hrss. The above plots are those corresponding to the low mass binary, while the plots below are those corresponding to the high mass binary. The results are displayed for pycWB with $\beta = 6$ and MESA with $ar = 500$ and $\beta = 2$.

from General Relativity. Conversely, highly massive binaries spend only a short time in the detector band and are mainly sensitive to the reconstruction of the merger and ring-down phases, which are especially important for testing the strong-field regime of General Relativity, higher modes, and quasi-normal modes.

We first focus on the low-mass binary, shown in the upper panels of Figure E.1. In this case MESA provides a visibly larger overlap with the injected waveform compared to the standard pycWB reconstruction. The histograms in the left panel show the overlap distribution obtained from all reconstructions, with black corresponding to pycWB and red to MESA. The dashed colored lines indicate the median values of the distributions, while the dotted lines represent the overlap between the injected waveform and the median reconstructed waveform over the ensemble. Since the latter is computed from the median reconstruction itself, it can systematically exceed the values characterizing the individual-event distributions.

For the low-mass system the histogram is clearly shifted toward higher overlap values for MESA, with an increase of approximately 1.1% in the median overlap. An even larger gain is observed for the median reconstruction, whose overlap increases by approximately 2.8%. The cumulative hrss plot in the right panel explains the origin of this improvement.

Most of the gain is accumulated during the first half of the signal, where MESA follows the injected cumulative hrss much more accurately than pycWB. In the later part of the waveform the two methods become more similar. The improvement therefore appears to originate primarily from a better reconstruction of the inspiral phase, which is characterized by a relatively low local SNR.

We now consider the high-mass binary, shown in the lower panels of Figure E.1. In this case the differences between the two methods are significantly smaller. The overlap histogram still shows a slight shift toward larger values for MESA, corresponding to an increase of approximately 0.5% in the median overlap. However, the cumulative hrss reconstruction does not exhibit a substantial improvement. Both methods reconstruct the injected hrss with comparable accuracy over the full duration of the signal.

The modest gain observed in the overlap may instead be related to the larger variability of the MESA reconstructions. In both cases the lower bound of the 90% confidence interval is similar between the two methods, but MESA occasionally reaches higher overlap values. This broader distribution may explain the slightly larger median overlap, despite the overall similarity of the two reconstructions.

Appendix F

Study of the eigenvector to highlight systematic errors

In this appendix we report the very first results obtained from the investigations of the NCM framework applied to GW250114 (now reported in 8.2.2). Despite being incorrect, we decided to report these results because they show the strength of the Network Covariance Matrix Framework.
As we will see in the appendix, there were two main sources of error in the reconstruction, accounting for 55% of the whole variance in the reconstruction. These two modes are related to a phase error and an amplitude error. By undergoing several runs and investigations, we discovered that the main phase error (accounting for 33% of the total variance!) was an artifact of the post-production analysis, that was not taking into account the sub-sample alignment of the waveform. This systematic was affecting our previous analysis and results. Interestingly, the Network Covariance Matrix was showing exactly were this systematic was, and made possible to find and fix it.
We think that these results can show how strong the NCM can be, and the importance of the interpretability of the eigenvectors in the context of our analysis.

The top panel of Figure F.1 shows the first eigenvector of the network and the direct sum of the single detector eigenvectors. For the first direction, the network eigenvector is almost perfectly aligned with the direct sum of the single detector eigenvectors, indicating that the relevant reconstruction is the same whether considering the single detectors separately or the network as a whole suggesting that correlation is not relevant for this reconstruction.

The physical origin of this direction is revealed by comparing the eigenvector with the reconstructed on-source waveform, as shown in the upper panel of Figure F.2. There, we can see that the first eigenvector represents a ($\pm 90°$) phase shift of the original waveform. This phase shift is the primary source of error in the reconstruction, accounting for 30% of the total variance. The bottom panel shows the $\pm 1\sigma$ confidence interval (gray band) compared with the first eigenvector with opposite signs. It is evident that positive and negative components along the first eigenvectors can explain a high variance of waveform reconstruction around the merger time. The third eigenvector, shown in the third panel of figure F.1 is again representative of a phase error, orthogonal to the previous one. This is the representation of the systematic error that we were including due to the "fractional" time shift of the waveforms. Thanks to this result, we have been able to identify this systematic

error and correct it.

The second panel of Figure F.1 represents the second eigenvector. When compared with the on-source reconstruction (central panel of F.2), this error is aligned with the reconstructed waveform itself, i.e. it represent the error in the estimate of the amplitude, as for the first eigenvector of Section 8.2.2. A main difference is that, in this first analysis, it was accounting for $\sim 25\%$ of the total variance, while in the correct results its impact decreased to $\sim 2\%$. As in the main text, the following eigenvector becomes of more difficult interpretation.

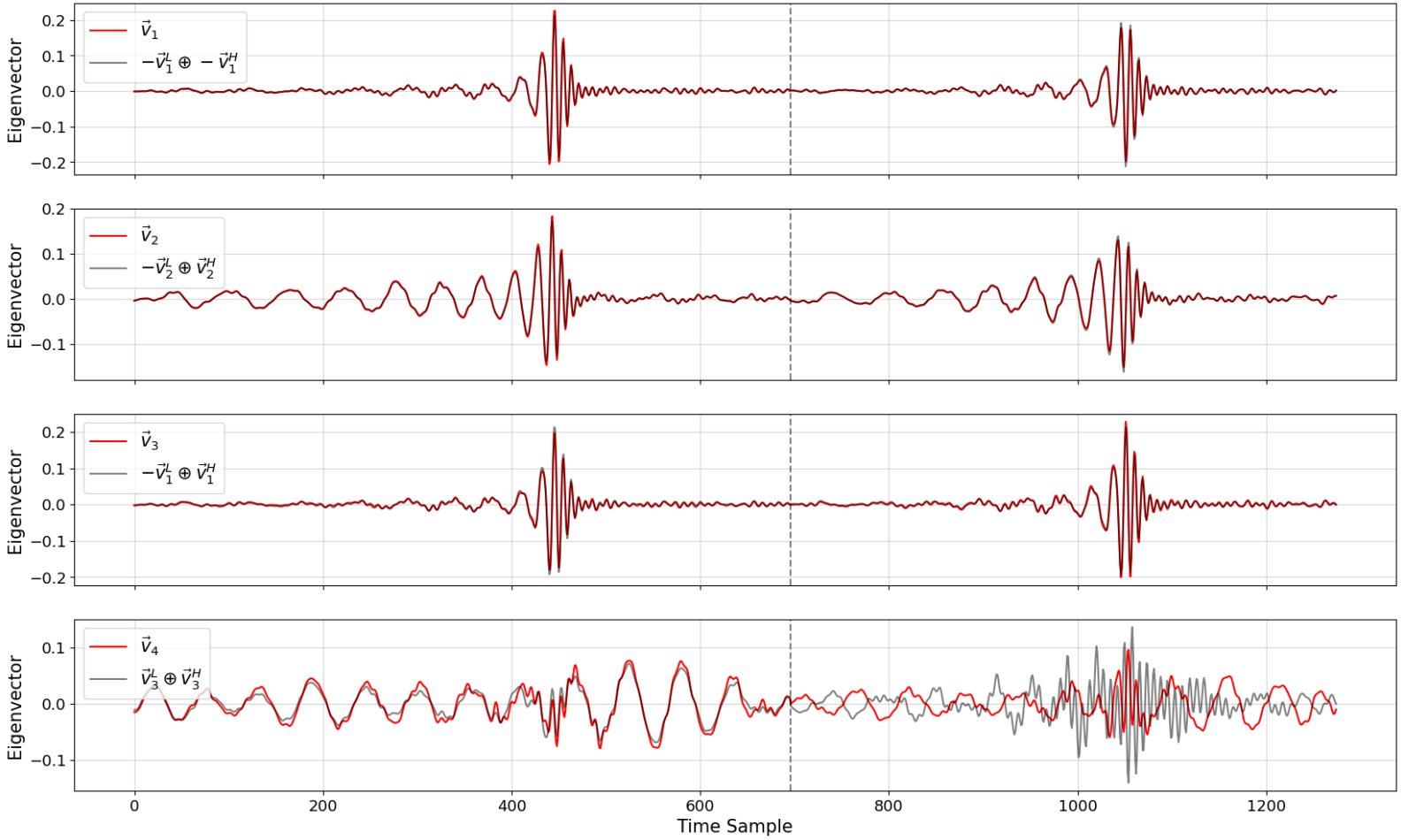


Figure F.1: First four eigenvectors of the network covariance matrix (red curves), together with the direct sum of the first four eigenvectors of the single detector reconstruction. The network eigenvectors are normalized to have unit norm, while the single detector eigenvectors are normalized to have unit norm in their respective subspace, so that they can be directly compared with the network eigenvectors.

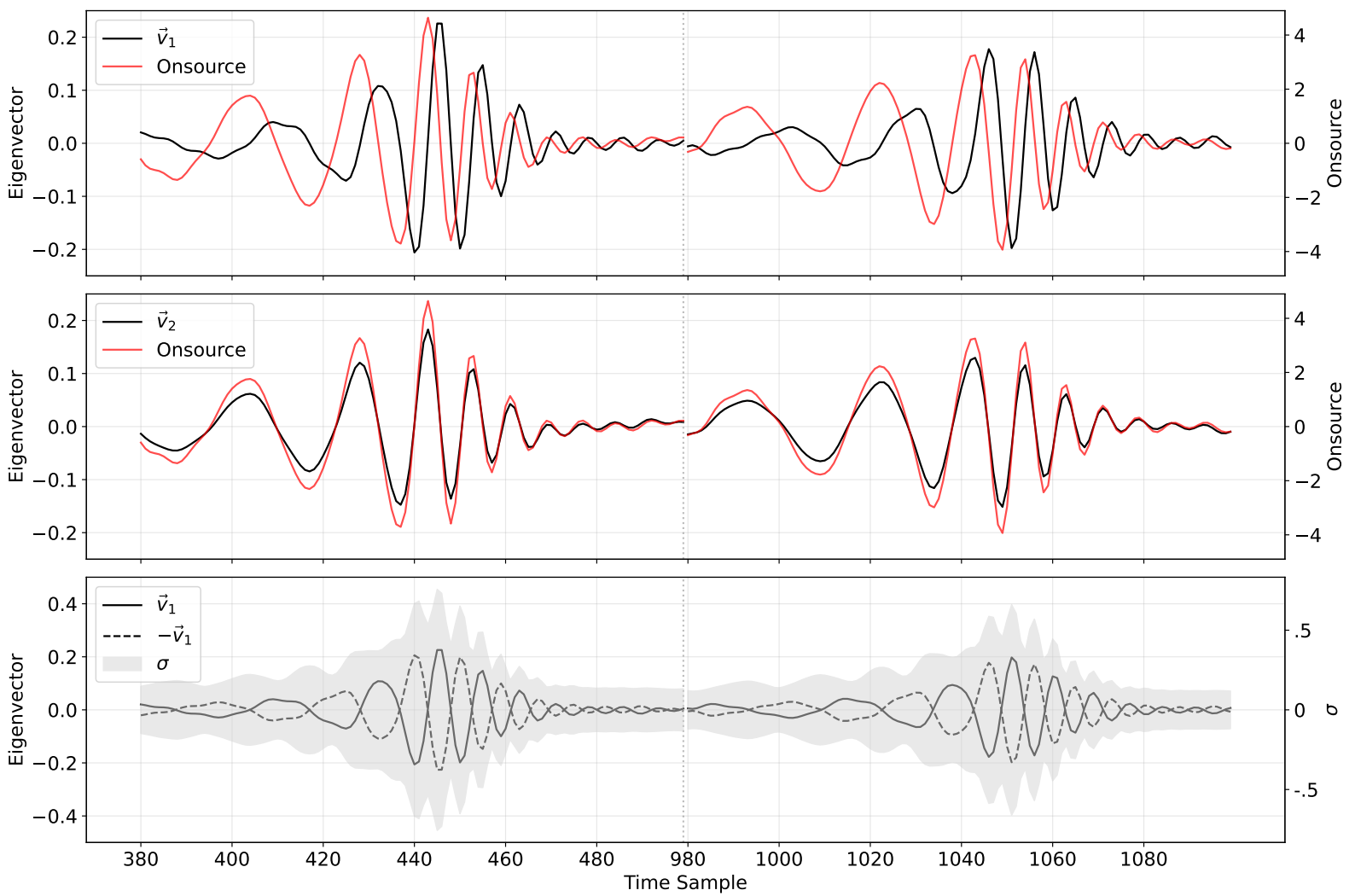


Figure F.2: Comparison of the first two eigenvectors with respect to the time-domain waveform reconstructed on-source by HL. The top panel shows the first eigenvector against the on-source reconstruction for the signal. The central panel shows the second eigenvector against the on-source reconstruction for the signal. Bottom panel shows plus or minus the first eigenvector against the $\pm 1\sigma$ confidence interval.

Appendix G

Posterior Samples

In this appendix we report the distribution of the relevant parameters for the signal $GW250114$ discussed in Section 2.3.2, The gray histogram represent the marginal distribution for the parameter distribution obtained from the posterior samples obtained with bilby [26]. The blue histogram represent the distribution of the parameters selected with the $\mathcal{M}^2$ statistic, as discussed in Section 2.3.2. The black dashed line represent the maximum a Posteriori estimate for the relative parameter while red dashed line represents the estimate obtained by minimizing the $\mathcal{M}^2$ statistic. For comparison, the Maximum Likelihood estimate, represented by the green dashed line, is also reported.

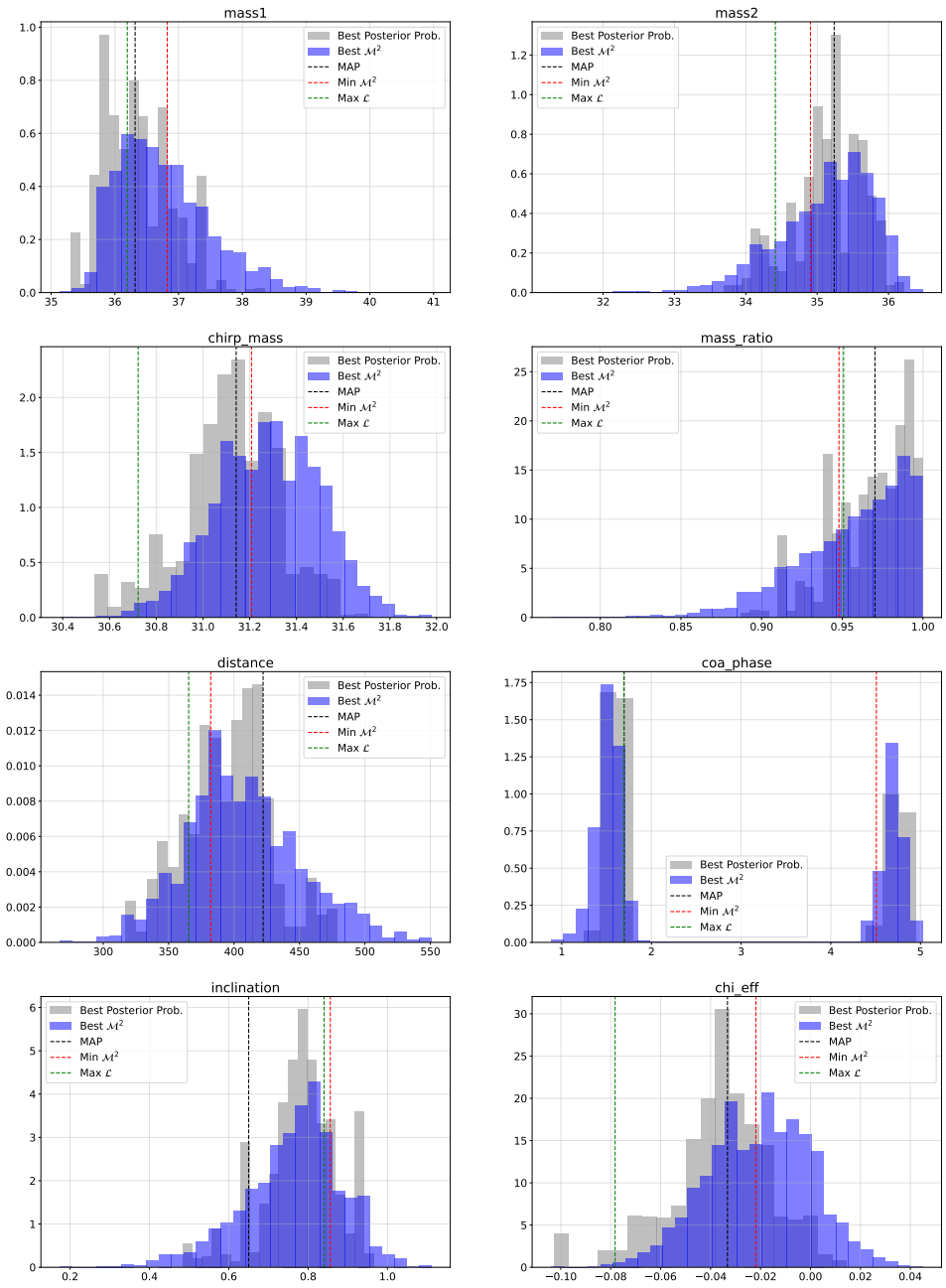


Figure G.1: Comparison of the posterior distributions of the parameters for the Posterior Estimate and the $\mathcal{M}^2$ statistics.

Bibliography

[1] J. Aasi et al. “Advanced LIGO”. In: *Class. Quant. Grav.* 32 (2015), p. 074001. DOI: 10.1088/0264-9381/32/7/074001. arXiv: 1411.4547 [gr-qc].

[2] Junaid Aasi et al. “Enhanced sensitivity of the LIGO gravitational wave detector by using squeezed states of light”. In: *Nature Photonics* 7.8 (2013), pp. 613–619.

[3] A. G. Abac et al. “GW250114: Testing Hawking’s Area Law and the Kerr Nature of Black Holes”. In: *Phys. Rev. Lett.* 135 (11 Sept. 2025), p. 111403. DOI: 10.1103/kw5g-d732. URL: https://link.aps.org/doi/10.1103/kw5g-d732.

[4] A. G. Abac et al. “Search for Gravitational Waves Emitted from SN 2023ixf”. In: *The Astrophysical Journal* 985.2 (May 2025), p. 183. DOI: 10.3847/1538-4357/adc681. URL: https://doi.org/10.3847/1538-4357/adc681.

[5] J. Abadie et al. “All-sky search for gravitational-wave bursts in the second joint LIGO-Virgo run”. In: *Phys. Rev. D* 85 (12 June 2012), p. 122007. DOI: 10.1103/PhysRevD.85.122007. URL: https://link.aps.org/doi/10.1103/PhysRevD.85.122007.

[6] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116 (6 Feb. 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061102.

[7] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116 (2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061102.

[8] B. P. Abbott et al. “Observing gravitational-wave transient GW150914 with minimal assumptions”. In: *Phys. Rev. X* 6 (2016), p. 041015. DOI: 10.1103/PhysRevX.6.041015. arXiv: 1606.04856 [gr-qc]. URL: https://arxiv.org/abs/1606.04856.

[9] B. P. Abbott et al. “Optically targeted search for gravitational waves emitted by core-collapse supernovae during the first and second observing runs of advanced LIGO and advanced Virgo”. In: *Phys. Rev. D* 101 (8 Apr. 2020), p. 084002. DOI: 10.1103/PhysRevD.101.084002. URL: https://link.aps.org/doi/10.1103/PhysRevD.101.084002.

[10] R. Abbott et al. “All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO and Advanced Virgo O3 data”. In: *Phys. Rev. D* 106 (10 Nov. 2022), p. 102008. DOI: 10.1103/PhysRevD.106.102008. URL: https://link.aps.org/doi/10.1103/PhysRevD.106.102008.

[11] R. Abbott et al. “All-sky, all-frequency directional search for persistent gravitational waves from Advanced LIGO’s and Advanced Virgo’s first three observing runs”. In: *Phys. Rev. D* 105 (12 June 2022), p. 122001. DOI: 10.1103/PhysRevD.105.122001. URL: https://link.aps.org/doi/10.1103/PhysRevD.105.122001.

[12] R. Abbott et al. “GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run”. In: *Phys. Rev. X* 13 (2023), p. 041039. DOI: 10.1103/PhysRevX.13.041039.

[13] R. Abbott et al. “Population of Merging Compact Binaries Inferred Using Gravitational Waves through GWTC-3”. In: *Phys. Rev. X* 13 (1 Mar. 2023), p. 011048. DOI: 10.1103/PhysRevX.13.011048. URL: https://link.aps.org/doi/10.1103/PhysRevX.13.011048.

[14] Ernazar Abdikamalov, Giulia Pagliaroli, and David Radice. “Gravitational Waves from Core-Collapse Supernovae”. In: (Oct. 2020). DOI: 10.1007/978-981-15-4702-7_21-1. arXiv: 2010.04356 [astro-ph.SR].

[15] J. G. Ables. “Maximum Entropy Spectral Analysis”. In: *Astronomy and Astrophysics Supplement* 15 (1974), pp. 383–393.

[16] F. Acernese et al. “Advanced Virgo: a second-generation interferometric gravitational wave detector”. In: *Class. Quant. Grav.* 32.2 (2015), p. 024001. DOI: 10.1088/0264-9381/32/2/024001. arXiv: 1408.3978 [gr-qc].

[17] F. Acernese et al. “Virgo detector characterization and data quality: results from the O3 run”. In: *Class. Quant. Grav.* 40 (2023), p. 185006. DOI: 10.1088/1361-6382/acd92d. arXiv: 2210.15633 [gr-qc]. URL: https://arxiv.org/abs/2210.15633.

[18] Fausto Acernese et al. “Frequency-Dependent Squeezed Vacuum Source for the Advanced Virgo Gravitational-Wave Detector”. In: *Physical Review Letters* 131.4 (2023), p. 041403.

[19] Scott M. Adams et al. “OBSERVING THE NEXT GALACTIC SUPERNOVA”. In: *The Astrophysical Journal* 778.2 (Nov. 2013), p. 164. DOI: 10.1088/0004-637X/778/2/164. URL: https://doi.org/10.1088/0004-637X/778/2/164.

[20] Hirotugu Akaike. “Statistical Predictor Identification”. In: *Selected Papers of Hirotugu Akaike*. Ed. by Emanuel Parzen, Kunio Tanabe, and Genshiro Kitagawa. New York, NY: Springer New York, 1998, pp. 137–151. ISBN: 978-1-4612-1694-0. DOI: 10.1007/978-1-4612-1694-0_11. URL: https://doi.org/10.1007/978-1-4612-1694-0_11.

[21] Bruce Allen et al. “FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries”. In: *Phys. Rev. D* 85 (12 June 2012), p. 122006. DOI: 10.1103/PhysRevD.85.122006. URL: https://link.aps.org/doi/10.1103/PhysRevD.85.122006.

[22] T. W. Anderson and D. A. Darling. “A Test of Goodness of Fit”. In: *Journal of the American Statistical Association* 49 (1954), pp. 765–769. DOI: 10.1080/01621459.1954.10501232.

[23] Danai Antonopoulou, Brynmor Haskell, and Cristóbal M Espinoza. “Pulsar glitches: observations and physical interpretation”. In: *Reports on Progress in Physics* 85.12 (Dec. 2022), p. 126901. DOI: 10.1088/1361-6633/ac9ced. URL: https://doi.org/10.1088/1361-6633/ac9ced.

[24] Nicolas Arnaud et al. "Coincidence and coherent data analysis methods for gravitational wave bursts in a network of interferometric detectors". In: *Phys. Rev. D* 68 (10 Nov. 2003), p. 102001. DOI: 10.1103/PhysRevD.68.102001. URL: https://link.aps.org/doi/10.1103/PhysRevD.68.102001.

[25] G Ashton and C Talbot. "Bilby-MCMC: an MCMC sampler for gravitational-wave inference". In: *Monthly Notices of the Royal Astronomical Society* 507.2 (Oct. 2021), pp. 2037–2051. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2236. eprint: https://academic.oup.com/mnras/article-pdf/507/2/2037/44923607/stab2236.pdf. URL: https://doi.org/10.1093/mnras/stab2236.

[26] G. Ashton, M. Hübner, P. D. Lasky, et al. *Bilby: A user-friendly Bayesian inference library for gravitational-wave astronomy*. 2019. DOI: 10.3847/1538-4365/ab06fc. URL: https://git.ligo.org/lscsoft/bilby.

[27] Yoichi Aso et al. "Interferometer design of the KAGRA gravitational wave detector". In: *Phys. Rev. D* 88 (4 Aug. 2013), p. 043007. DOI: 10.1103/PhysRevD.88.043007. URL: https://link.aps.org/doi/10.1103/PhysRevD.88.043007.

[28] T. E. Barnard. *The maximum entropy spectrum and the Burg technique. Technical report no. 1: Advanced signal processing*. NASA STI/Recon Technical Report N. 1975.

[29] S. Bini et al. "An autoencoder neural network integrated into gravitational-wave burst searches to improve the rejection of noise transients". In: *Classical and Quantum Gravity* 40 (2023), p. 135008. DOI: 10.1088/1361-6382/acd981. URL: https://dx.doi.org/10.1088/1361-6382/acd981.

[30] Hendrik W. Bode. *Network Analysis and Feedback Amplifier Design*. R. E. Krieger Pub. Co., 1975. Chap. 14.

[31] J. P. Burg. *Maximum Entropy Spectral Analysis*. Stanford University, 1975. URL: https://books.google.it/books?id=Xug_AAAAIAAJ.

[32] M. Cabero et al. "Blip glitches in Advanced LIGO data". In: *Classical and Quantum Gravity* 36 (2019). DOI: 10.1088/1361-6382/ab2e14.

[33] Gravitational Wave Open Science Center. *GWOSC: Gravitational Wave Open Science Center*. 2023. URL: https://www.gw-openscience.org/.

[34] Katerina Chatziioannou et al. "Morphology-independent test of the mixed polarization content of transient gravitational wave signals". In: *Phys. Rev. D* 104 (4 Aug. 2021), p. 044005. DOI: 10.1103/PhysRevD.104.044005. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.044005.

[35] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[36] Demetrios Christodoulou. "Nonlinear nature of gravitation and gravitational-wave experiments". In: *Phys. Rev. Lett.* 67 (12 Sept. 1991), pp. 1486–1489. DOI: 10.1103/PhysRevLett.67.1486. URL: https://link.aps.org/doi/10.1103/PhysRevLett.67.1486.

[37] *coherent WaveBurst web-site, https://gwburst.gitlab.io*. Accessed: 2025-03-10. URL: https://gwburst.gitlab.io.

[38] The LIGO Scientific Collaboration et al. *All-sky search for long-duration gravitational-wave transients in the first part of the fourth LIGO-Virgo-KAGRA Observing run.* 2025. arXiv: 2507.12282 [gr-qc]. URL: https://arxiv.org/abs/2507.12282.

[39] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *Constraints on gravitational waves from the 2024 Vela pulsar glitch.* 2026. arXiv: 2512.17990 [gr-qc]. URL: https://arxiv.org/abs/2512.17990.

[40] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *GWTC-4.0: Tests of General Relativity. II. Parameterized Tests.* 2026. arXiv: 2603.19020 [gr-qc]. URL: https://arxiv.org/abs/2603.19020.

[41] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *GWTC-4.0: Updating the Gravitational-Wave Transient Catalog with Observations from the First Part of the Fourth LIGO-Virgo-KAGRA Observing Run.* 2025. arXiv: 2508.18082 [gr-qc]. URL: https://arxiv.org/abs/2508.18082.

[42] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *GWTC-5.0: Methods for Identifying and Characterizing Gravitational-wave Transients.* 2026. arXiv: 2605.27224 [gr-qc]. URL: https://arxiv.org/abs/2605.27224.

[43] The LIGO Scientific Collaboration, the Virgo Collaboration, and the KAGRA Collaboration. *GWTC-5.0: Observations from the Second Part of the Fourth LIGO-Virgo-KAGRA Observing Run and Updates to the Gravitational-Wave Transient Catalog.* 2026. arXiv: 2605.27225 [gr-qc]. URL: https://arxiv.org/abs/2605.27225.

[44] The LIGO Scientific Collaboration et al. *GWTC-4.0: Tests of General Relativity. I. Overview and General Tests.* 2026. arXiv: 2603.19019 [gr-qc]. URL: https://arxiv.org/abs/2603.19019.

[45] The LIGO Scientific Collaboration et al. *Open Data from LIGO, Virgo, and KAGRA through the First Part of the Fourth Observing Run.* 2025. arXiv: 2508.18079 [gr-qc]. URL: https://arxiv.org/abs/2508.18079.

[46] The Virgo Collaboration. *Advanced Virgo Plus Phase I - Design Report.* Tech. rep. The Virgo Collaboration, 2019.

[47] N. J. Cornish. "Time-frequency analysis of gravitational wave data". In: *Phys. Rev. D* 102 (2020), p. 124038. DOI: 10.1103/PhysRevD.102.124038. URL: https://link.aps.org/doi/10.1103/PhysRevD.102.124038.

[48] Neil J Cornish and Tyson B Littenberg. "Bayeswave: Bayesian inference for gravitational wave bursts and instrument glitches". In: *Classical and Quantum Gravity* 32.13 (2015), p. 135012. ISSN: 1361-6382. DOI: 10.1088/0264-9381/32/13/135012. URL: http://dx.doi.org/10.1088/0264-9381/32/13/135012.

[49] Neil J. Cornish. "Time-frequency analysis of gravitational wave data". In: *Phys. Rev. D* 102 (12 Dec. 2020), p. 124038. DOI: 10.1103/PhysRevD.102.124038. URL: https://link.aps.org/doi/10.1103/PhysRevD.102.124038.

[50] Curt Cutler et al. "The last three minutes: Issues in gravitational-wave measurements of coalescing compact binaries". In: *Phys. Rev. Lett.* 70 (20 May 1993), pp. 2984–2987. DOI: 10.1103/PhysRevLett.70.2984. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.2984.

[51] Derek Davis et al. "LIGO detector characterization in the second and third observing runs". In: *Classical and Quantum Gravity* 38.13 (2021), p. 135014.

[52] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[53] Walter Del Pozzo and John Veitch. *raynest: Nested sampling algorithm parallelized through Ray*. GitHub repository. 2015. URL: https://github.com/wdpozzo/raynest.

[54] scikit-learn developers. *SKLearn*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html.

[55] SciPy developers. *scipy*. URL: https://docs.scipy.org/doc/.

[56] M. Drago et al. "coherent WaveBurst, a pipeline for unmodeled gravitational-wave data analysis". In: *SoftwareX* 14 (2021), p. 100678. DOI: 10.1016/j.softx.2021.100678. URL: https://www.sciencedirect.com/science/article/pii/S2352711021000236.

[57] Matthew C. Edwards et al. "Bayesian semiparametric power spectral density estimation with applications in gravitational wave data analysis". In: *Phys. Rev. D* 92 (6 Sept. 2015), p. 064011. DOI: 10.1103/PhysRevD.92.064011. URL: https://link.aps.org/doi/10.1103/PhysRevD.92.064011.

[58] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Book. 1993.

[59] D. Gabor. "Theory of communication". In: *J. Inst. Electr. Engineering* (1946), pp. 429–457.

[60] Bhooshan ando others Gadre. "Detectability of eccentric binary black holes with matched filtering and unmodeled pipelines during the third observing run of LIGO-Virgo-KAGRA". In: *Phys. Rev. D* 110 (4 Aug. 2024), p. 044013. DOI: 10.1103/PhysRevD.110.044013. URL: https://link.aps.org/doi/10.1103/PhysRevD.110.044013.

[61] Edward Higson et al. "Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation". In: *Statistics and Computing* 29 (2019), pp. 891–913. DOI: 10.1007/s11222-018-9844-0. URL: https://doi.org/10.1007/s11222-018-9844-0.

[62] K. Hirata et al. "Observation of a neutrino burst from the supernova SN1987A". In: *Phys. Rev. Lett.* 58 (14 Apr. 1987), pp. 1490–1493. DOI: 10.1103/PhysRevLett.58.1490. URL: https://link.aps.org/doi/10.1103/PhysRevLett.58.1490.

[63] Wynn C. G. Ho et al. "Gravitational waves from transient neutron star f-mode oscillations". In: *Phys. Rev. D* 101 (10 May 2020), p. 103009. DOI: 10.1103/PhysRevD.101.103009. URL: https://link.aps.org/doi/10.1103/PhysRevD.101.103009.

[64] G Hobbs et al. "The International Pulsar Timing Array project: using pulsars as a gravitational wave detector". In: *Classical and Quantum Gravity* 27.8 (Apr. 2010), p. 084013. DOI: 10.1088/0264-9381/27/8/084013. URL: https://doi.org/10.1088/0264-9381/27/8/084013.

[65] G. Hobbs and S. Dai. "Pulsar timing array projects". In: *National Science Review* 4.5 (2017), pp. 707–717. DOI: 10.1093/nsr/nwx118.

[66] R. A. Hulse and J. H. Taylor. "Discovery of a pulsar in a binary system." In: *apjl* 195 (1975), pp. L51–L53. DOI: 10.1086/181708.

[67] Hans-Thomas Janka. "Explosion Mechanisms of Core-Collapse Supernovae". In: *Annual Review of Nuclear and Particle Physics* 62 (2012), pp. 407–451. DOI: 10.1146/annurev-nucl-102711-094901.

[68] E. T. Jaynes. “Information Theory and Statistical Mechanics”. In: *Physical Review* 106 (1957), pp. 620–630.

[69] E. T. Jaynes and G. L. Bretthorst. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[70] Steven M Kay. *Modern spectral estimation*. Pearson Education India, 1988.

[71] S. Klimenko et al. “A coherent method for detection of gravitational wave bursts”. In: *Classical and Quantum Gravity* 25 (2008), p. 114029. DOI: 10.1088/0264-9381/25/11/114029. URL: https://dx.doi.org/10.1088/0264-9381/25/11/114029.

[72] S. Klimenko et al. “Constraint likelihood analysis for a network of gravitational wave detectors”. In: *Phys. Rev. D* 72 (2005), p. 122002. DOI: 10.1103/PhysRevD.72.122002. URL: https://link.aps.org/doi/10.1103/PhysRevD.72.122002.

[73] S. Klimenko et al. “Localization of gravitational wave sources with networks of advanced detectors”. In: *Phys. Rev. D* 83 (2011), p. 102001. DOI: 10.1103/PhysRevD.83.102001. URL: https://link.aps.org/doi/10.1103/PhysRevD.83.102001.

[74] S. Klimenko et al. “Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors”. In: *Phys. Rev. D* 93 (2016), p. 042004. DOI: 10.1103/PhysRevD.93.042004. URL: https://link.aps.org/doi/10.1103/PhysRevD.93.042004.

[75] Sergey Koposov et al. *joshspeagle/dynesty: v3.0.0*. Version 3.0.0. 2025. DOI: 10.5281/zenodo.17268284. URL: https://github.com/joshspeagle/dynesty.

[76] M. Kramer et al. “Strong-Field Gravity Tests with the Double Pulsar”. In: *Phys. Rev. X* 11 (4 Dec. 2021), p. 041050. DOI: 10.1103/PhysRevX.11.041050.

[77] Jack YL Kwok et al. “Investigation of the effects of non-Gaussian noise transients and their mitigation in parameterized gravitational-wave tests of general relativity”. In: *Physical Review D* 105.2 (2022), p. 024066.

[78] N. Levinson. “The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction”. In: *Journal of Mathematics and Physics* 25 (1946), pp. 261–278.

[79] Tyson B. Littenberg and Neil J. Cornish. “Bayesian inference for spectral estimation of gravitational wave detector noise”. In: *Phys. Rev. D* 91 (8 Apr. 2015), p. 084034. DOI: 10.1103/PhysRevD.91.084034. URL: https://link.aps.org/doi/10.1103/PhysRevD.91.084034.

[80] F. T. Liu et al. “Isolation Forest”. In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[81] F. Lusvardi et al. *Sky Localization of Gravitational Wave Transients with Generic Morphology Using LIGO-Virgo Data*. 2026.

[82] Ronaldas Macas et al. “Impact of noise transients on low latency gravitational-wave event localization”. In: *Physical Review D* 105.10 (2022), p. 103021.

[83] D. M. Macleod et al. “GWpy: A Python package for gravitational-wave astrophysics”. In: *SoftwareX* 13 (2021), p. 100657. ISSN: 2352-7110. DOI: 10.1016/j.softx.2021.100657. URL: https://www.sciencedirect.com/science/article/pii/S2352711021000029.

[84] M. Maggiore. *Gravitational Waves. Vol. 1: Theory and Experiments*. Oxford University Press, 2007. DOI: 10.1093/acprof:oso/9780198570745.001.0001.

[85] M. Maggiore. *Gravitational Waves. Vol. 2: Astrophysics and Cosmology*. Oxford University Press, 2018.

[86] P C Mahalanobis. "On the generalised distance in statistics". In: *Proceedings National Institute of Science. India* (1936). URL: http://ir.isical.ac.in/dspace/handle/1/1268.

[87] A Martini et al. In: 43.5 (Mar. 2026), p. 055016. DOI: 10.1088/1361-6382/ae4717. URL: https://doi.org/10.1088/1361-6382/ae4717.

[88] A. Martini et al. "Maximum entropy spectral analysis: an application to gravitational waves data analysis". In: *Eur. Phys. J. C* 84 (2024), p. 1023. DOI: 10.1140/epjc/s10052-024-13400-6. URL: https://doi.org/10.1140/epjc/s10052-024-13400-6.

[89] A. Martini et al. *memspectrum: A Fast Maximum Entropy Spectral Analysis Package*. Version 1.3.0. 2024. URL: https://pypi.org/project/memspectrum/.

[90] Y. Meyer. *Ondelettes et opérateurs: Ondelettes. I*. Actualités mathématiques. Editions Hermann, 1990. ISBN: 9782705661250.

[91] Andrea Miani et al. "Constraints on the amplitude of gravitational wave echoes from black hole ringdown using minimal assumptions". In: *Phys. Rev. D* 108 (6 Sept. 2023), p. 064018. DOI: 10.1103/PhysRevD.108.064018. URL: https://link.aps.org/doi/10.1103/PhysRevD.108.064018.

[92] V. Necula et al. "Transient analysis with fast Wilson-Daubechies time-frequency transform". In: *Journal of Physics: Conference Series* 363 (2012), p. 012032. DOI: 10.1088/1742-6596/363/1/012032. URL: https://dx.doi.org/10.1088/1742-6596/363/1/012032.

[93] A. Nitz, I. W. Harry, D. A. Brown, et al. *PyCBC software*. 2024. URL: https://github.com/gwastro/pycbc.

[94] Chris Pankow et al. "Mitigation of the instrumental noise transient in gravitational-wave data surrounding GW170817". In: *Physical Review D* 98.8 (2018), p. 084016.

[95] P. C. Peters and J. Mathews. "Gravitational Radiation from Point Masses in a Keplerian Orbit". In: *Phys. Rev.* 131 (1 July 1963), pp. 435–440. DOI: 10.1103/PhysRev.131.435. URL: https://link.aps.org/doi/10.1103/PhysRev.131.435.

[96] Konstantin A. Postnov and Lev R. Yungelson. "The Evolution of Compact Binary Star Systems". In: *Living Reviews in Relativity* 17.1 (May 2014), p. 3. DOI: 10.12942/lrr-2014-3.

[97] Geraint Pratten et al. "Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes". In: *Phys. Rev. D* 103 (10 May 2021), p. 104056. DOI: 10.1103/PhysRevD.103.104056. URL: https://link.aps.org/doi/10.1103/PhysRevD.103.104056.

[98] M. Pürrer et al. *IMRPhenomD: Phenomenological waveform model*. Astrophysics Source Code Library, record ascl:2307.019. 2023. ascl: 2307.019.

[99] Antoni Ramos-Buades et al. "Impact of eccentricity on the gravitational-wave searches for binary black holes: High mass case". In: *Phys. Rev. D* 102 (4 Aug. 2020), p. 043005. DOI: 10.1103/PhysRevD.102.043005. URL: https://link.aps.org/doi/10.1103/PhysRevD.102.043005.

[100] Philip Relton et al. "Addressing the challenges of detecting time-overlapping compact binary coalescences". In: *Physical Review D* 106.10 (2022), p. 104045.

[101] K. Riles. "Gravitational waves: Sources, detectors and searches". In: *Progress in Particle and Nuclear Physics* 68 (2013), pp. 1–54. ISSN: 0146-6410. DOI: https://doi.org/10.1016/j.ppnp.2012.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0146641012001093.

[102] Florent Robinet et al. "Omicron: A tool to characterize transient noise in gravitational-wave detectors". In: *SoftwareX* 12 (2020), p. 100620. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2020.100620. URL: https://www.sciencedirect.com/science/article/pii/S2352711020303332.

[103] C. Röver et al. "Modelling coloured residual noise in gravitational-wave signal processing". In: *Classical and Quantum Gravity* 28 (2011). DOI: 10.1088/0264-9381/28/1/015010.

[104] Shio others Sakon. "Template bank for compact binary mergers in the fourth observing run of Advanced LIGO, Advanced Virgo, and KAGRA". In: *Phys. Rev. D* 109 (4 Feb. 2024), p. 044066. DOI: 10.1103/PhysRevD.109.044066. URL: https://link.aps.org/doi/10.1103/PhysRevD.109.044066.

[105] F. Salemi et al. "Wider look at the gravitational-wave transients from GWTC-1 using an unmodeled reconstruction method". In: *Phys. Rev. D* 100 (2019), p. 042003. DOI: 10.1103/PhysRevD.100.042003. URL: https://link.aps.org/doi/10.1103/PhysRevD.100.042003.

[106] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27 (1948), pp. 379–423.

[107] John Skilling. "Nested sampling for general Bayesian computation". In: *Bayesian Analysis* 1 (Dec. 2006), pp. 833–860. DOI: 10.1214/06-BA127.

[108] S Soni et al. "LIGO Detector Characterization in the first half of the fourth Observing run". In: *Classical and Quantum Gravity* 42.8 (Apr. 2025), p. 085016. DOI: 10.1088/1361-6382/adc4b6. URL: https://doi.org/10.1088/1361-6382/adc4b6.

[109] Patrick J Sutton et al. "X-Pipeline: an analysis package for autonomous gravitational-wave burst searches". In: *New Journal of Physics* 12.5 (May 2010), p. 053034. DOI: 10.1088/1367-2630/12/5/053034. URL: https://doi.org/10.1088/1367-2630/12/5/053034.

[110] M. J. Szczepańczyk et al. "Search for gravitational-wave bursts in the third Advanced LIGO-Virgo run with coherent WaveBurst enhanced by machine learning". In: *Phys. Rev. D* 107 (2023), p. 062002. DOI: 10.1103/PhysRevD.107.062002. URL: https://link.aps.org/doi/10.1103/PhysRevD.107.062002.

[111] Eric Thrane and Colm Talbot. "An introduction to Bayesian inference in gravitational-wave astronomy: Parameter estimation, model selection, and hierarchical models". In: *Publications of the Astronomical Society of Australia* 36 (2019), e010. DOI: 10.1017/pasa.2019.2.

[112] J. Timmer and M. Koenig. "On generating power law noise". In: *Astronomy and Astrophysics* 300 (1995), p. 707.

[113] V. Tiwari et al. "Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors". In: *Classical and Quantum Gravity* 32 (2015), p. 165014. DOI: 10.1088/0264-9381/32/16/165014. URL: https://dx.doi.org/10.1088/0264-9381/32/16/165014.

[114] T. J. Ulrych and T. N. Bishop. "Maximum entropy spectral analysis and autoregressive decomposition". In: *Reviews of Geophysics* 13 (1975), pp. 183–200. DOI: 10.1029/RG013i001p00183. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/RG013i001p00183.

[115] S. A. Usman et al. "The PyCBC search for gravitational waves from compact binary coalescence". In: *Classical and Quantum Gravity* 33.21 (Oct. 2016), p. 215004. DOI: 10.1088/0264-9381/33/21/215004. URL: https://doi.org/10.1088/0264-9381/33/21/215004.

[116] David Vartanyan et al. "Gravitational-wave signature of core-collapse supernovae". In: *Phys. Rev. D* 107 (10 May 2023), p. 103015. DOI: 10.1103/PhysRevD.107.103015.

[117] J. Veitch et al. "Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library". In: *Phys. Rev. D* 91 (4 Feb. 2015), p. 042003. DOI: 10.1103/PhysRevD.91.042003. URL: https://link.aps.org/doi/10.1103/PhysRevD.91.042003.

[118] J. Veitch and A. Vecchio. "Bayesian approach to the follow-up of candidate gravitational wave signals". In: *Phys. Rev. D* 78 (2 July 2008), p. 022001. DOI: 10.1103/PhysRevD.78.022001. URL: https://link.aps.org/doi/10.1103/PhysRevD.78.022001.

[119] K. Vos. *A Fast Implementation of Burg's Algorithm*. 2013. URL: https://opus-codec.org/docs/vos_fastburg.pdf.

[120] Qingwen Wang et al. "Echoes from quantum black holes". In: *Phys. Rev. D* 101 (2 Jan. 2020), p. 024031. DOI: 10.1103/PhysRevD.101.024031. URL: https://link.aps.org/doi/10.1103/PhysRevD.101.024031.

[121] Peter Welch. "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.

[122] Y. Xu. *pyCWB*. URL: https://git.ligo.org/yumeng.xu/pycwb.

[123] Y. Xu et al. "PycWB: A user-friendly, Modular, and python-based framework for gravitational wave unmodelled search". In: *SoftwareX* 26 (2024), p. 101639. DOI: 10.1016/j.softx.2024.101639. URL: https://www.sciencedirect.com/science/article/pii/S2352711024000104.

[124] Yumeng Xu et al. "Enhancing gravitational wave parameter estimation with nonlinear memory: Breaking the distance-inclination degeneracy". In: *Phys. Rev. D* 109 (12 June 2024), p. 123034. DOI: 10.1103/PhysRevD.109.123034. URL: https://link.aps.org/doi/10.1103/PhysRevD.109.123034.

[125] M Zevin et al. In: *Classical and Quantum Gravity* 34.6 (Feb. 2017), p. 064003. DOI: 10.1088/1361-6382/aa5cea. URL: https://doi.org/10.1088/1361-6382/aa5cea.

List of Figures

List of Tables