



Searches for Continuous Gravitational Waves: Sensitivity Estimation and Deep Learning as a Novel Search Method

Von der Fakultät für Mathematik und Physik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

Doktor der Naturwissenschaften
Dr.rer.nat.

genehmigte Dissertation
von

M.Sc. Christoph Dreißigacker
geboren am 18.02.1990 in Hildesheim

2020

Referentin: Dr. Maria Alessandra Papa
Korreferenten: Dr. Christopher Messenger
Dr. Alicia Magdalena Sintes
Dr. Reinhard Prix

Promotionskommission: Prof. Dr. Domenico Giulini
Prof. Dr. Benno Willke
Dr. Christopher Messenger
Dr. Reinhard Prix

Tag der Promotion: 25.09.2020

Abstract

The first direct detections of gravitational waves from merging black holes and neutron stars started the era of gravitational-wave astronomy. Since then, observing merging compact objects has become routine. Other exciting sources still remain undetected.

Rapidly-rotating neutron stars are predicted to emit weak, long-lasting quasi-monochromatic waves called continuous gravitational waves (CWs). In the current detector generation, advanced LIGO and Virgo, various noise sources create far more signal output than a potential CW signal. CW data analysis tries to overcome the weakness of the signals by integrating over long stretches of data. Analyzing large amounts of data usually corresponds to large computing cost. For that reason, CW searches for signals from unknown neutron stars are limited in their sensitivity by computational cost.

This thesis is concerned with estimating and improving the sensitivity of continuous gravitational wave searches. The first main research work presented in this thesis is a new sensitivity estimator that can swiftly and accurately predict the sensitivity of a CW search before it is started. This makes optimizing the search algorithms and therefore improving the sensitivity easier. The accuracy of the estimator is studied by applying it to many different CW searches. The work is expanded with an extensive sensitivity review of past CW searches by calculating their sensitivity depth.

The second main part of this thesis is the development of a new CW search method based on deep neural networks (DNNs). DNNs are extremely fast once trained and therefore might present an interesting possibility of circumventing the computational limitations and creating a more sensitive CW search. In this thesis such a DNN CW search is developed first as a single-detector search for signals from all over the sky and then expanded to a multi-detector all-sky search and to directed multi-detector searches for signals from a single position in the sky. The DNNs' performance is compared to coherent matched-filtering searches in terms of detection probability at fixed false-alarm level first on idealized Gaussian noise and then on realistic LIGO detector noise. This thesis finds that the DNNs show a lot of potential: For short timespans of about one day the networks only lose a few percent in sensitivity depth compared to coherent matched-filtering. For longer timespans the networks' performance gradually deteriorates making further research necessary. As an outlook to future research, this thesis proposes the combination of short-timespan network outputs, similar to semi-coherent matched-filtering, as a DNN search method over longer timespans.

Keywords: continuous gravitational waves, data analysis, sensitivity estimation, deep learning

Kurzfassung

Die ersten direkten Detektionen von Gravitationswellen von verschmelzenden Schwarzen Löchern und Neutronensternen haben die Ära der Gravitationswellenastronomie eingeläutet. Seitdem ist die Beobachtung von verschmelzenden kompakten Objekten zur Routine geworden. Andere interessante Quellen von Gravitationswellen sind jedoch noch unentdeckt.

Schnell rotierende Neutronensterne können schwache, langanhaltende quasi-monochromatische Wellen aussenden, genannt Kontinuierliche Gravitationswellen (engl.: *continuous waves* CWs). In der aktuellen Detektorgeneration, advanced LIGO und Virgo, wird mehr Detektoroutput durch diverse Rauschquellen erzeugt als durch potenzielle CW Signale. Die Datenanalyse für CWs versucht die Schwäche der Signale zu überwinden, indem die Daten über lange Zeitspannen integriert werden. Große Mengen von Daten zu analysieren ist jedoch für gewöhnlich mit großen Ansprüchen an die Rechenleistung verbunden. Deshalb sind Suchen nach CWs von unbekanntem Neutronensternen in ihrer Empfindlichkeit limitiert durch die begrenzt vorhandene Rechenleistung.

Diese Doktorarbeit beschäftigt sich mit dem Abschätzen und Verbessern der Empfindlichkeit von Suchen nach Kontinuierlichen Gravitationswellen. Das erste Hauptforschungsergebnis dieser Arbeit ist ein neuartiger Abschätzer, der die Empfindlichkeit einer CW-Suche schnell und genau vorhersagen kann bevor die Suche gestartet wird. Dies vereinfacht die Verbesserung der Suchalgorithmen und kann deshalb zu empfindlicheren Suchen führen. Die Genauigkeit des Abschätzers wird anhand von vielen verschiedenen CW-Suchen untersucht. Die Untersuchung wird ergänzt durch eine ausführliche Studie der Empfindlichkeit von vergangenen CW-Suchen. Dazu wird deren Empfindlichkeit in die gemeinsame Größe der Empfindlichkeitstiefe (engl.: *sensitivity depth*) umgerechnet.

Das zweite Hauptforschungsergebnis dieser Dissertation ist eine neuartige CW-Suchmethode mit Hilfe von tiefen neuronalen Netzwerken (engl.: *deep neural networks*, DNNs). Fertigt trainierte DNNs können extrem schnell angewendet werden und stellen deshalb eine interessante Art und Weise dar, wie möglicherweise mit der Limitierung durch fehlende Rechenleistung umgegangen und eine empfindlichere Suche konstruiert werden kann. In dieser Arbeit wird eine solche DNN nutzende CW-Suchmethode präsentiert: zuerst als Suche mit Daten von einem einzigen Detektor nach Signalen vom gesamten Himmel und dann erweitert zu Suchen mit Daten von mehreren Detektoren nach Signalen vom gesamten Himmel oder nach Signalen von einer speziellen Himmelsposition. Die Leistungsfähigkeit der DNNs wird dabei verglichen mit kohärenten Optimalfiltermethoden im Hinblick auf ihre Detektionswahrscheinlichkeit bei festem Fehlalarmniveau. Diese Arbeit zeigt das diesbezüglich große Potential von DNNs: Bei der Analyse von kurzen Zeitspannen von etwa einem Tag verliert das Netzwerk nur wenige Prozent in Empfindlichkeitstiefe gegenüber kohärenten Optimalfiltermethoden. Für längere Zeitspannen nimmt die Leistungsfähigkeit der Netzwerke im Vergleich jedoch nach und nach ab. An dieser Stelle wird deshalb weitere Forschungsarbeit benötigt um die Leistungsfähigkeit der DNNs zu verbessern. Ein Ansatz, der in dieser Arbeit für zukünftige Forschung vorgeschlagen wird, ist die Kombination von Ergebnissen, die Netzwerke auf kurzen Zeitspannen erreicht haben, als Ergebnis für längere Zeitspannen zu nutzen. Dieser Ansatz ist ähnlich zum semi-kohärenten Optimalfilter, der in klassischen CW-Suchen benutzt wird.

Schlagerworte: Kontinuierliche Gravitationswellen, Datenanalyse, Empfindlichkeitsabschätzung, Deep Learning, Neuronale Netzwerke

Contents

Abstract	ii
Kurzfassung	iii
Contents	v
1. Introduction	1
2. Chapter Descriptions and Authorship Clarifications	3
3. Gravitational Waves	5
3.1. Wave Solutions in Linearized General Relativity	5
3.2. Effect of a Gravitational Wave on Free-Falling Particles	8
3.3. Gravitational-Wave Detectors	9
3.4. Astrophysical Sources of Gravitational Waves	12
3.4.1. Compact Binary Coalescence	13
3.4.2. Continuous Gravitational Waves	14
4. Continuous-Wave Data Analysis	19
4.1. Source and Solar System Barycenter Reference Frames	19
4.2. Detector Response	21
4.3. CW Signal Model	22
4.4. Matched-Filtering Searches	23
4.4.1. The Detection Problem: Frequentist and Bayesian approach	23
4.4.2. Detector Data	25
4.4.3. \mathcal{F} -statistic	27
4.4.4. Templates and Computing Cost	28
4.4.5. Semi-coherent Searches and Follow-ups	29
5. Deep Learning	31
5.1. Deep Learning and Deep Neural Networks	31
5.2. Neurons, Layers, Networks	33
5.3. The Training Loop of a Neural Network	35
5.3.1. Loss and Gradient Descent	35
5.3.2. Parameter Update	39
5.4. Main Challenges in Deep-Learning	40
5.4.1. Preparing Data	40
5.4.2. Overfitting and Underfitting	41
5.4.3. Network Architecture and Hyperparameters	43
5.4.4. Vanishing Gradient Problem	44
5.5. Advanced Concepts	45
5.5.1. Convolutional Layer	45
5.5.2. Pooling Layers	46
5.5.3. Normalization and Batch Normalization	47
5.5.4. Residual Connections and Inception Modules	47

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches	51
6.0. Abstract	51
6.1. Introduction	51
6.2. \mathcal{F} -statistic-based Search Methods	54
6.2.1. Signal Model	54
6.2.2. Coherent \mathcal{F} -statistic	55
6.2.3. Semi-coherent \mathcal{F} -statistic Methods	56
6.2.4. Mismatch and Template Banks	57
6.2.5. Sensitivity Depth	58
6.3. Sensitivity Estimate	58
6.3.1. Frequentist Upper Limits	58
6.3.2. Approximating Wide-parameter-space Statistics	58
6.3.3. StackSlide- \mathcal{F} Sensitivity	59
6.3.4. Multi-stage StackSlide- \mathcal{F} Sensitivity	61
6.3.5. Hough- \mathcal{F} Sensitivity	62
6.3.6. Bayesian Upper Limits	62
6.3.7. Numerical Implementation	63
6.4. Determining Frequentist Upper Limits	63
6.4.1. Sigmoid p_{det} Interpolation	64
6.4.2. Piecewise-linear Threshold Interpolation	66
6.5. Comparing Estimates Against Simulated Upper Limits	67
6.5.1. Example: S6-AllSky-StackSlide- \mathcal{F} Search	67
6.5.2. Example: Multi-directed O1-MD-StackSlide- \mathcal{F}	69
6.6. Comparing Estimates Against Measured Upper limits	69
6.6.1. General Remarks and Caveats	70
6.6.2. All-sky Searches	71
6.6.3. Directed and Narrow-band Searches	73
6.6.4. Searches for Neutron Stars in Binaries	76
6.6.5. Targeted Searches for Known Pulsars	76
6.7. Discussion	78
6.8. Appendix A: Details on Referenced CW Searches	78
6.8.1. General Remarks	78
6.8.2. All-sky Searches, see Table 6.1	79
6.8.3. Directed Searches, see Tables 6.2, 6.3	83
6.8.4. Searches for Neutron Stars in Binary Systems, see Table 6.4	85
6.8.5. Targeted Searches, see Table 6.5	87
6.9. Appendix B: CW Signal Model and \mathcal{F} -statistic	89
6.10. Appendix C: Definition of the Geometric Factor R^2	90
6.11. Appendix D: Distribution of \mathcal{F} -statistic Maximized over Correlated Templates	90
7. Deep-Learning Continuous Gravitational Waves	93
7.0. Abstract	93
7.1. Introduction	93
7.2. Comparison Test Benchmarks	94
7.2.1. Benchmark Definitions	94
7.2.2. WEAVE Matched-filtering Sensitivity	95
7.3. Deep-Learning CWs	97
7.3.1. Network Architecture	97
7.3.2. DNN Training and Validation	98
7.4. Testing DNN Performance	100
7.4.1. Verifying Detection Probabilities	100
7.4.2. Generalization in Frequency f	101

7.4.3. Generalization in Spin-down \dot{f}	101
7.4.4. Generalization in Signal Strength	102
7.4.5. Timing	103
7.5. Discussion	105
8. Deep-Learning Continuous Gravitational Waves: Multiple detectors and realistic noise	107
8.0. Abstract	107
8.1. Introduction	107
8.2. Comparison Test Benchmarks	108
8.3. Deep-Learning CWs	110
8.3.1. Finding a Network Architecture	110
8.3.2. DNN Training and Validation	111
8.4. Characterizing DNN Performance on Gaussian Noise	112
8.4.1. Detection Probabilities at Fixed False Alarm	112
8.4.2. Generalization	113
8.5. Testing Network Performance on Real Data	118
8.5.1. Gaussian Noise with Data Gaps	119
8.5.2. Performance on Real Detector Data	120
8.6. Discussion	120
9. Outlook	123
Appendix A. Bibliography	125
Appendix B. List of Tables	143
Appendix C. List of Figures	145
Appendix D. List of Publications	151
Appendix E. Curriculum Vitae	153
Appendix F. Acknowledgements	155

1. Introduction

Since the formulation of the two main theories of physics, quantum mechanics and general relativity, physicists wondered how to combine these two theories. Few experiments can reveal a combined effect as quantum mechanics is mostly relevant for small length scales while gravity, i.e. general relativity, is most relevant for the large scale structures of the universe.

Compact astronomical objects such as black holes, neutron stars and white-dwarfs do need both theories to be accurately described. And while classical electromagnetic observations can shed some light on the structure and prevalence of these objects, gravitational waves can add a lot of additional information. This was shown when the LIGO-Virgo collaboration announced the first direct detection of gravitational waves from black holes [1] and from neutron stars [2]. While the first detections of merging black holes were the starting point of gravitational-wave astronomy and gave us insight into these fascinating objects, the binary neutron star merger GW170817 showed us the first glimpse of the exciting future of multi-messenger astronomy from gravitational waves, electromagnetic waves and neutrinos [3]. The multi-messenger detection has shown us e.g. a direct connection of binary-neutron-star mergers and short gamma-ray bursts and kilonova explosions [3], it allowed an independent measurement of the expansion of the universe [4] and established neutron-star mergers as a major source of heavy elements in the universe [3, 5, 6].

Gravitational-wave signals from compact binary mergers, however, were only the beginning of gravitational-wave astronomy. Other exciting types of gravitational waves and sources still have to be detected. In this thesis we will focus mainly on the predicted gravitational waves from rapidly-rotating neutron stars. These continuous gravitational waves (CWs) are long-lasting quasi-monochromatic waves that are expected to be several orders of magnitude weaker in amplitude than binary-merger signals. A detection of a continuous wave could bring insight to the structure and prevalence of neutron stars especially as it could potentially also be a multi-messenger signal if the neutron star is also visible via electromagnetic radiation (e.g. a pulsar). Continuous gravitational waves can also be emitted by other sources like white-dwarf binaries and boson clouds around black holes, further increasing their significance and potentially allowing interesting tests of the main theories of physics.

CW data analysis tries to overcome the weakness of the CW signals by analyzing large amounts of data, integrating over long timespans to separate the periodic signal from the random noise. The necessary amount of data is in fact so large that the searches for signals from unknown neutron stars are limited by the availability of computing power despite running for several months on large computing clusters or powerful volunteer computing projects such as Einstein@Home [7].

The long run-times of the searches pose additional challenges to the analysis on their own. One problem, studied in Ch. 6 of this thesis, is the fast and accurate estimation of the sensitivity of such a search. This is often necessary prior to the search for planning and properly setting up these long-duration searches, but it can also be used for quick sanity checks of results. The typical sensitivity calculation for a finished CW search is done with additional large-scale searches with artificially injected signals, making it expensive and not easily repeatable. Therefore, a sensitivity estimator can indirectly help in the main task of continuous-wave data analysis: improving the sensitivity until a detection is made.

Deep neural networks (DNNs) have recently experienced a large breakthrough, achieving super-human abilities in specialized tasks like image recognition, speech recognition, visual object detection as well as in more fundamental applications like drug research and genomics [8]. They

1. Introduction

have also been applied to gravitational-wave related research to classify disturbances, for searches and parameter estimation of binary merger signals, long-transient signals and follow-ups to continuous wave searches [9–27].

Typically, these networks are difficult and time-consuming to train properly but easy and fast to apply to data. In CW searches the data is usually split in many smaller segments. Therefore, a network could potentially be trained once but applied many times, making the actual search very fast. Thus, the low computing cost of neural networks could ultimately lead to a more sensitive search. In Chs. 7 and 8 of this thesis we developed a deep-learning based CW search method and compared it to previous search methods.

2. Chapter Descriptions and Authorship Clarifications

Chapters 3 to 5 briefly introduce fundamental concepts necessary for the comprehension of the following chapters. These concepts are partially also discussed in the introduction sections of the published work presented in chapters 6,7 and 8 but are here presented in a broader context. Chapter 3 gives a general introduction to gravitational waves in the context of the theory of general relativity, gravitational-wave detectors and astrophysical sources. Chapter 4 presents continuous-wave data analysis as used for matched-filtering based searches. Chapter 5 establishes the main concepts of deep learning and deep neural networks. These three chapters were entirely written by the author with some help in proof reading.

Chapter 6 is a reprint of the work published as Dreissigacker, Prix, and Wette [28]. It presents a fast and accurate sensitivity estimator for some important types of searches for continuous gravitational waves. This estimator is an improvement of a previous analytic approach by Wette [29]. It also contains an extensive summary of most types of continuous gravitational-wave searches. For this review part, the sensitivity in the form of published upper limits of past CW searches was converted to a common quantity called sensitivity depth. The idea for this work was provided by R. Prix and K. Wette. The development of the sensitivity estimator was done by the author in close collaboration with R. Prix and with some help of K. Wette who developed the analytic predecessor. The initial draft of the text was written by the author and revisions were performed by R. Prix. The final editing was done in close cooperation with R. Prix.

Chapter 7 is a reprint of the work published as Dreissigacker et al. [30]. It presents a novel approach for a search for continuous gravitational waves using deep learning. The chapter is a proof-of-principle study of this approach, establishing benchmark results for a single-detector all-sky deep-learning searches and comparing the results to coherent matched-filtering sensitivities. It shows that deep neural networks can be trained to detect continuous gravitational waves and illustrates some remarkable abilities to generalize its sensitivity to signals that were not presented to the network during training. As final result it concludes that for short timespans of data the networks can come close to the matched-filtering sensitivity while it lacks further behind for longer timespans. The idea for this work was provided by C. Messenger and Ruining Zhao. The software was initially developed by R. Sharma in collaboration with R. Prix and the author and later by the author in collaboration with R. Prix. The final network architecture search and the training of the networks was done by the author. The matched-filtering searches for the comparison results were performed by R. Prix. The text was written by the author under the close guidance of R. Prix.

Chapter 8 is a reprint of the work published as Dreissigacker and Prix [31]. The chapter is a continuation of the work presented in Ch. 7. It expands on the previous study by using simulated strain data from two detectors simultaneously and by training networks to perform a search for continuous waves from a fixed sky position. Similar to the previous work it shows that deep neural networks can detect CW signals in short timespans of data almost as well as coherent matched-filtering searches but they fall short for longer timespans. The chapter goes on to show that the neural networks perform well even for signals with parameters that were not contained in the training set. The chapter also presents the first test of network performance on real LIGO data, establishing that non-Gaussian noise disturbances significantly reduce the sensitivity. The DNN software development, network architecture optimization and training were performed by the author. The matched-filtering comparison software was mainly developed by R. Prix and adapted and run for the benchmark specifications by the author. The text was written by the author with some feedback from R. Prix.

Chapter 9 gives some concluding remarks and an outlook by the author.

3. Gravitational Waves

In the 1860s James Clerk Maxwell discovered that light can be described as an electromagnetic wave following what we now call the Maxwell equations [32]. This theory turned out to be very accurate and other researchers, such as Oliver Heaviside [33] and Henry Poincaré [34], started wondering if a gravitational equivalent to electromagnetic waves exists. After Albert Einstein completed his theory of general relativity in 1915 [35], the question of whether gravitational waves existed became controversial. Einstein himself changed his mind multiple times, at some point simply concluding that he does not know if gravitational waves exist [36]. However, over time the understanding of Einstein’s theory grew and by the time the first binary pulsar was discovered in 1974, it was possible to show that its orbit decayed consistent with the energy loss due to gravitational-wave emissions predicted by general relativity [37]. A further wide-spread belief that gravitational waves can never be detected directly due to their weakness, was proven false in 2015 when the LIGO collaboration directly detected a signal from two merging black holes [1]. While this discovery and the following binary-merger observations started the era of gravitational-wave astronomy, other types of gravitational waves have still to be detected.

In this chapter we will very briefly introduce the basic equations of Einstein’s theory of relativity and show how we can derive a wave equation in the linear approximation. For this purpose we will make use of index notation. Greek spacetime indices take values from 0 to 4, while roman spatial indices take values from 1 to 3. This chapter also uses the Einstein summation convention, i.e. expressions with repeating indices contain an implicit sum over this index.

We will go on discussing the effect a gravitational wave has and how it is measured in a laser-interferometric gravitational wave detector. And finally we will discuss gravitational-wave sources: first in the context of the first detections and then we will focus on continuous gravitational waves as the main topic of this work.

3.1. Wave Solutions in Linearized General Relativity

In general relativity gravitation is no longer described as a force but instead as curvature of the spacetime manifold. The Einstein equations (for Gravitational constant and speed of light $G = c = 1$ and without cosmological constant $\Lambda = 0$)

$$G_{\mu\nu} = 8\pi T_{\mu\nu} \tag{3.1}$$

describe how matter and energy, in the form of the stress-energy tensor $T_{\mu\nu}$, curves spacetime given by the Einstein tensor $G_{\mu\nu}$. On the other hand, the geodesic equation of general relativity describes the motion of a particle as a parametrized path $x^\mu(\tau)$

$$\frac{d^2 x^\mu}{d\tau^2} + \Gamma^\mu_{\nu\rho} \frac{dx^\nu}{d\tau} \frac{dx^\rho}{d\tau} = F^\mu. \tag{3.2}$$

It tells us that acceleration is either caused by non-gravitational forces F^μ or by curvature of spacetime given by the Christoffel symbol $\Gamma^\mu_{\nu\rho}$. John Archibald Wheeler described this very briefly and poignantly [38]: “Space tells matter how to move; matter tells space how to curve”¹.

¹He later corrected it to say “spacetime” instead of “space”.

3. Gravitational Waves

Curvature in these equations can be characterized by a number of different quantities, that are all functions of the metric of the spacetime manifold $g_{\mu\nu}$ and its derivatives with respect to the coordinate directions $\partial_\mu := \frac{\partial}{\partial x^\mu}$:

$$G_{\mu\nu} := R_{\mu\nu} - \frac{R}{2}g_{\mu\nu} \quad (3.3)$$

$$R := R^\mu{}_\mu \quad (3.4)$$

$$R_{\mu\nu} := R^\alpha{}_{\mu\alpha\nu} \quad (3.5)$$

$$R^\alpha{}_{\mu\beta\nu} := \partial_\beta \Gamma^\alpha{}_{\mu\nu} - \partial_\nu \Gamma^\alpha{}_{\mu\beta} + \Gamma^\alpha{}_{\sigma\beta} \Gamma^\sigma{}_{\mu\nu} - \Gamma^\alpha{}_{\sigma\nu} \Gamma^\sigma{}_{\mu\beta} \quad (3.6)$$

$$\Gamma^\alpha{}_{\mu\nu} := \frac{1}{2}g^{\alpha\rho} (\partial_\mu g_{\rho\nu} + \partial_\nu g_{\rho\mu} - \partial_\rho g_{\mu\nu}). \quad (3.7)$$

The metric of the spacetime manifold $g_{\mu\nu}$ can be used to raise and lower indices as

$$x_\mu = g_{\mu\nu}x^\nu \quad \text{and} \quad x^\mu = g^{\mu\nu}x_\nu \quad (3.8)$$

where $g^{\mu\nu}$ are the components of the inverse metric.

Equations (3.3)-(3.7) show that the Einstein equations contain many non-linear terms, making them a non-linear system of partial differential equations. As such the Einstein equations have proven to be notoriously difficult to solve. Nonetheless, some analytic solutions for the Einstein equations have been found, most notably the class of vacuum solutions describing the external field of idealized astrophysical objects such as the Schwarzschild and Kerr solution. These solutions are most relevant in the strong gravity regime, e.g. to describe black holes.

Gravitational waves, on the other hand, are only small gravitational disturbances of spacetime. A spacetime with very weak gravitation can be approximated in a suitable coordinate system by

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \quad (3.9)$$

where η is the flat (gravitation-free) metric from special relativity and h with $|h_{\mu\nu}| \ll 1$ is the small disturbance caused by gravity [39]. With this approximation we can safely ignore h^2 terms and therefore can raise and lower indices with the flat metric η :

$$h^{\mu\nu} := \eta^{\mu\rho}\eta^{\nu\sigma}h_{\rho\sigma} \quad (3.10)$$

If we now define the trace-reversed tensor

$$\bar{h}^{\mu\nu} := h^{\mu\nu} - \frac{h}{2}\eta^{\mu\nu} \quad (3.11)$$

and use some of the gauge freedoms contained in the Einstein equations to require the *Lorenz gauge*

$$\partial_\nu \bar{h}^{\mu\nu} = 0, \quad (3.12)$$

we can rewrite the Einstein tensor of eq.(3.3) as

$$G_{\mu\nu} = -\frac{1}{2}\eta^{\alpha\beta}\partial_\alpha\partial_\beta\bar{h}_{\mu\nu}. \quad (3.13)$$

Note now that $\square := \eta^{\alpha\beta}\partial_\alpha\partial_\beta$ is in fact the definition of the *wave operator* or *d'Alembert-operator* in flat space. Hence, the Einstein equations (3.1) become a system of inhomogeneous wave equations

$$\square\bar{h}_{\mu\nu} = -16\pi T_{\mu\nu} \quad (3.14)$$

with source term $-16\pi T_{\mu\nu}$.

If we consider the equation far away from any relevant source, where $T_{\mu\nu} = 0$, we get the homogeneous wave equation. These equations are known to have *plane wave* solutions

$$\bar{h}_{\mu\nu} = H_{\mu\nu}e^{i\eta_{\alpha\beta}k^\alpha x^\beta + i\varphi} \quad (3.15)$$

with the phase φ and the wave vector $(k^\alpha) = (\omega, \vec{k})$, which gives the spatial direction of propagation \vec{k} and the angular frequency of the wave ω . This also implies a propagation speed of the wave of $1 = c$, the speed of light. It is also well known (see e.g. [39]) that any solution to this wave equation can be described as superposition of these plane waves. Therefore, we can continue by considering a plane wave.

While we already introduced four additional gauge equations for the Lorenz gauge, this leaves a gauge freedom of changing coordinates x_α by a vector ξ_α which fulfills the wave equation $\square\xi_\alpha = 0$. This freedom can be used to impose the transverse-traceless (“TT”) gauge [39]

$$H^\alpha{}_\alpha = 0 \qquad H_{\alpha\beta}U^\beta = 0 \qquad (3.16)$$

where we can choose the vector $U^\alpha = \delta^\alpha_0$ as the four-velocity of an object resting in the chosen coordinate system. This leads to $\bar{h}_{\mu\nu}$ in the transverse-traceless projection:

$$\bar{h}_{\mu\nu}^{TT} = h_{\mu\nu}^{TT}. \qquad (3.17)$$

For a gravitational wave propagating in z -direction $(k^\alpha) = (\omega, 0, 0, \omega)$ we get the amplitude tensor

$$H_{\mu\nu}^{TT} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A_+ & A_\times & 0 \\ 0 & A_\times & -A_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad (3.18)$$

with only two remaining degrees of freedom A_+, A_\times . These are called the “plus” and “cross” *polarizations* of gravitational waves. In analogy to electromagnetism a superposition of two waves, one of each (linear) polarization, is generally elliptically polarized. Thus, we can write the spatial part of an elliptically-polarized gravitational-wave signal h^{ij} as linear combination of the two polarization basis tensors [40]

$$e_+^{ij} = \hat{x}^i\hat{x}^j - \hat{y}^i\hat{y}^j \qquad (3.19)$$

$$e_\times^{ij} = \hat{x}^i\hat{y}^j + \hat{x}^j\hat{y}^i. \qquad (3.20)$$

given by coordinate unit vectors \hat{x} and \hat{y} . The gravitaitonal-wave tensor is then

$$h^{ij}(\tau) = h_\times(\tau)e_\times^{ij} + h_+(\tau)e_+^{ij}, \qquad (3.21)$$

parametrized in the source time τ with

$$h_+(\tau, z) = A_+ \cos(\omega(\tau - z) + \varphi_+) \qquad (3.22)$$

$$h_\times(\tau, z) = A_\times \sin(\omega(\tau - z) + \varphi_\times), \qquad (3.23)$$

the angular frequency ω and the phases $\varphi_+, \varphi_\times$.

The vacuum solutions to the linearized Einstein equations describe the propagation of a gravitational wave far from the source. In order to study how a gravitational wave is generated, we can use the *quadrupole formalism*: In the “local wave zone”, where the propagation of the wave is no longer significantly influenced by the source field, but still even less influenced by the rest of the universe [41], the gravitational field can be approximated with a multipole expansion. The lowest non-vanishing multipole order of the gravitational field is the quadrupole moment of the mass distribution. For a source that is moving slowly and that is small in size compared to the wavelength of the emitted waves, the quadrupole moment can be expressed as the Newtonian moment of inertia with the trace removed:

$$I_{ij} = \int \rho(x) \left[x_i x_j - \frac{1}{3} r^2 \delta_{ij} \right] d^3x \qquad (3.24)$$

3. Gravitational Waves

with the mass density $\rho(x)$ and r the distance to the center of the source. The gravitational-wave tensor in the traceless-transverse (TT) gauge can then be written as

$$h_{ij}^{TT}(\tau) = \frac{2}{r} \left[\ddot{I}_{ij}(\tau - r) \right]^{TT} \quad (3.25)$$

where TT indicates the transverse-traceless projection and the dots denote time derivatives [42].

The quadrupole formula (3.25) shows that spherically symmetric and rotating axisymmetric mass distributions do not emit gravitational waves as they do not possess a time-dependent mass quadrupole. However, small deviations from the symmetry would cause gravitational radiation.

3.2. Effect of a Gravitational Wave on Free-Falling Particles

In the last section we established that the linearized Einstein equations do predict gravitational waves. In order to verify this prediction in an experiment it is necessary to study which measurable effects a gravitational wave causes. As gravitational waves are fluctuations in the spacetime metric, the most direct effect should be a measurable distance change perpendicular to the direction of propagation (cf. eq. (3.18)).

If we look at the geodesic equation (3.2) for a free particle at rest and in the TT gauge for this frame, i.e. the vector U^μ of eq. (3.16) is the initial four-velocity of the particle:

$$\frac{d}{d\tau} U^\mu + \Gamma^\mu_{\nu\rho} U^\nu U^\rho = 0. \quad (3.26)$$

We can now calculate the acceleration for a particle at rest

$$\left. \frac{d}{d\tau} U^\mu \right|_{U^i=0} = -\Gamma^\mu_{00} = -\frac{1}{2} \eta^{\mu\nu} (2\partial_0 h_{\nu 0} - \partial_\nu h_{00}). \quad (3.27)$$

But we know from eq. (3.18) that in the TT-gauge all $h_{\alpha 0}$ vanish. Hence free particles are not accelerated and remain at the same coordinate positions [39].

However, in general relativity the proper distance between two coordinate positions can change if the spacetime metric changes: Let's assume two nearby particles are at $(x, y, z) = (0, 0, 0)$ and $(x, y, z) = (\delta x, 0, 0)$. The proper distance of the two particles for a gravitational wave propagating in z -direction can then be calculated as [39]:

$$\begin{aligned} l_x &:= \int_{P_0}^{P_1} ds \\ &= \int_{P_0}^{P_1} \sqrt{g_{\mu\nu} dx^\mu dx^\nu} \\ &= \int_0^{\delta x} dx \sqrt{g_{xx}} \\ &= \int_0^{\delta x} dx \sqrt{1 + h_{xx}^{TT}} \\ &= \int_0^{\delta x} dx \left(1 + \frac{1}{2} h_{xx}^{TT} + \mathcal{O}(h^2) \right) \\ &= \left(1 + \frac{1}{2} h_{xx}^{TT} \right) \delta x + \mathcal{O}(h^2) \end{aligned} \quad (3.28)$$

where we assumed in the last step that h_{xx}^{TT} is constant over δx . In our wave solution (3.15) h_{xx}^{TT} of course varies over time and hence the proper distance varies over time and is proportional to the amplitude of the wave. We can also derive the same expression if the particles are arranged on the y or z axis. As $h_{yy}^{TT} = -h_{xx}^{TT}$ the y -direction is always compressed if the x -direction is stretched and vice versa. On the other hand we have $h_{zz}^{TT} = 0$ and therefore we will not have

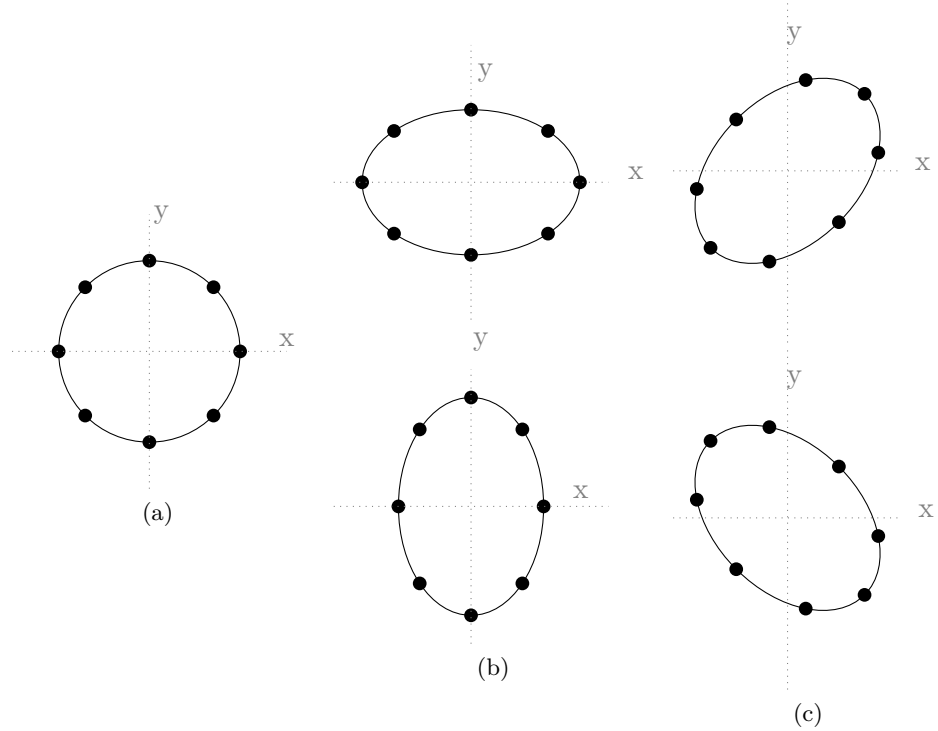


Figure 3.1.: A circle of free-falling particles in the x-y-plane (a) is distorted (b) by a +-polarized and (c) by a \times -polarized gravitational wave passing the setup in z-direction. Over time the effect periodically transitions from the top distortion via the circle to the bottom distortion and back.

any effect in the direction of propagation of the wave. Only perpendicularly space is periodically stretched and compressed as illustrated in Fig. 3.1.

Note also that if we have three particles at $(0, 0, 0)$, $(L, 0, 0)$ and $(0, L, 0)$, the difference ΔL in length between the proper distance along the x-axis l_x and the proper distance along the y-axis l_y is proportional to the initial distance L , we find:

$$\frac{\Delta L}{L} := \frac{l_x - l_y}{L} = h_{xx}^{TT} \quad (3.29)$$

This property will be very important in the next section on the detection of gravitational waves as the dimensionless components h_{ij}^{TT} of a gravitational wave are typically smaller than 10^{-21} and a larger detector magnifies the effect [39].

3.3. Gravitational-Wave Detectors

The first direct detection of gravitational waves happened in 2015 [1] by the LIGO collaboration by operating two large laser-interferometric detectors, the Laser Interferometer Gravitational-Wave Observatory (LIGO). A laser-interferometric gravitational-wave detector is based on the Michelson-interferometer. In these interferometers light is sent out by a laser towards a beam splitter. In the beam splitter the beam is split into two beams propagating perpendicular to each other along the two detector arms. At the end of these arms they are reflected back towards the beam splitter. At the beam splitter the two arm beams interfere and form two new beams, one back towards the laser and one perpendicular to it. At the latter the beam is directed to a photodetector, which measures the incoming photons. This principle is illustrated in Fig. 3.2.

3. Gravitational Waves

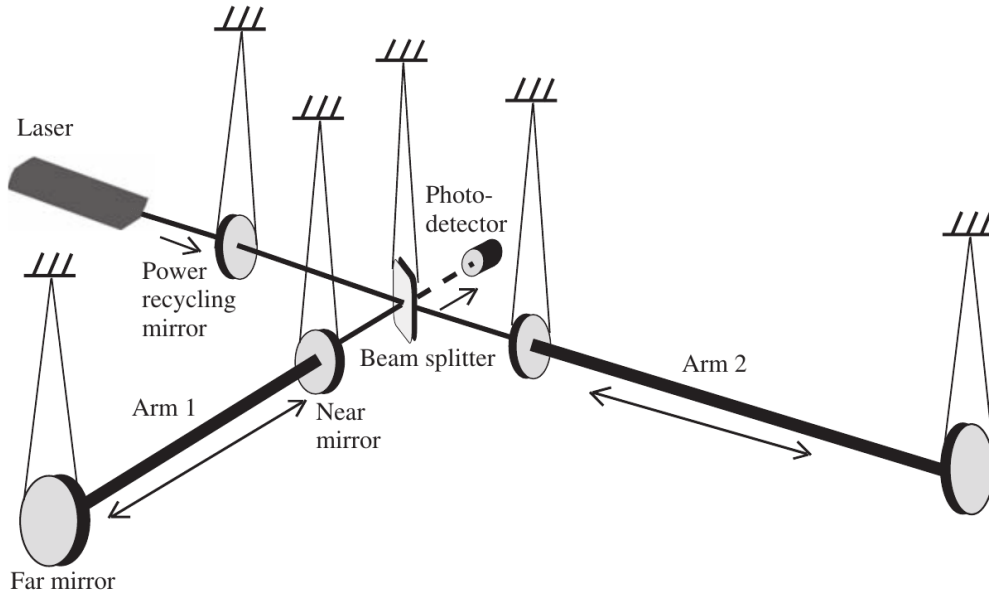


Figure 3.2.: Sketch of the configuration of a laser-interferometric detector like LIGO or Virgo: a Michelson interferometer, where all mirrors and the beam splitter are suspended as isolation against ground motion. Additionally, mirrors in the arms are used to form Fabry-Pérot cavities in order to increase the effective arm length. The power recycling mirror forms a cavity with the entire interferometer. It reflects the light which normally returns to the laser back into the interferometer increasing the power (Credit: Fig. 9.3 in [39]).

The power of received light at the photodetector is very sensitive to phase-shifts between the two arm beams. If for example a gravitational wave propagates such that one arm is compressed and one is stretched, this leads to different light travel times through the arms which causes a phase shift between the two beams altering the output on the photodetector. If we assume now that the mirrors in this detector are freely floating we can use the generalized eq. (3.29):

$$\Delta L = Lh \quad (3.30)$$

to notice that the length change ΔL is proportional to the strain amplitude h and the arm length of the detector L .

Therefore, the two main arms of the Michelson interferometer are usually chosen as long as possible. On Earth that lead to the construction of the 4 km-arm-length LIGO detectors [43, 44] and the 3 km Virgo detector [45, 46]. The smaller GEO600 detector [47] with 600 m arm length was not yet sensitive enough to confidently pick up any of the measured signals. Future ground-based detector designs are planned with even longer arms: The Einstein Telescope [48] is supposed to have 10 km long arms and the Cosmic Explorer design study [49] suggests up to 40 km arm length. Even longer arm lengths are planned for space-based missions. From 100 – 1000 km in the DECIGO proposal [50] to 2.5 Gm in the LISA proposal [51]. However, in all these detectors non-gravitational-wave related phase-shifts can be induced, making the output noisy and restricting these detectors to different sensitive frequency ranges.

The noise in the LIGO and Virgo detectors is not always fully explained, but most parts are understood and can be categorized in five categories [44] (see Fig. 3.3):

1. *Quantum noise* consists of two main parts:
 - at lower frequencies the *radiation pressure noise* is caused by the fluctuation in the number of photons hitting and displacing the optical components.

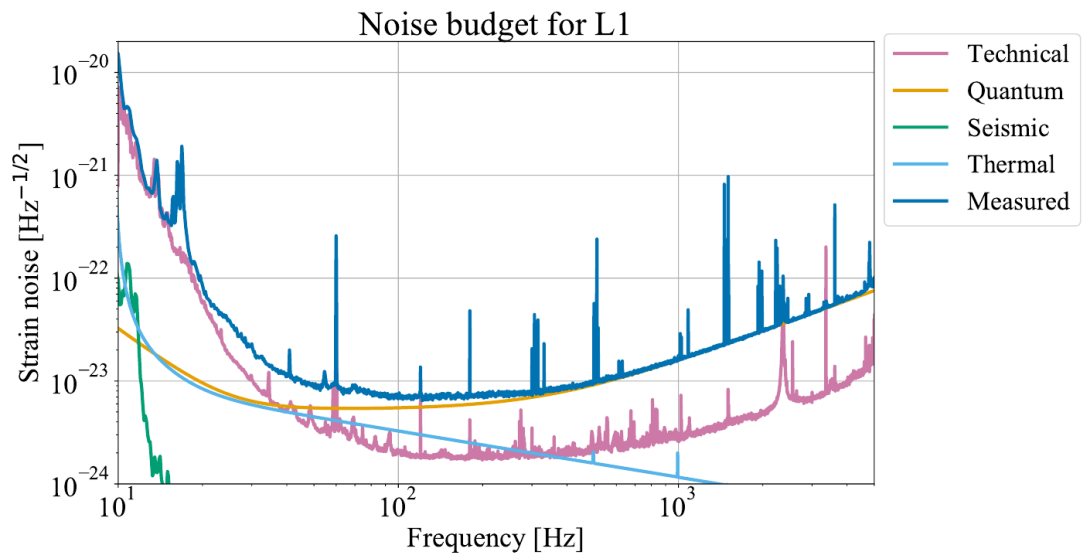


Figure 3.3.: O2 noise budget of the LIGO Livingston detector (L1). The measured equivalent noise spectrum (blue) and the main explainable contributions: Technical, quantum, seismic and thermal noise. The narrow peaks are called lines. Some of them have known causes, e.g. 60 Hz is the power grid frequency in the US, but many also have unknown causes (Credit: Fig. 3 of [52]).

- at higher frequencies the *shot noise* is caused by randomness in photon arrivals at the photodetector.
2. *Seismic noise* is caused by the non-static nature of the Earth. Earthquakes, tides, winds and waves as well as human activity cause vibrations of the ground, large enough to occlude a gravitational-wave signal.
 3. *Gravity gradient noise* (not shown in Fig. 3.3) is caused by seismic waves producing temporary density disturbances in the Earth close to the detectors. This causes fluctuations in the Earth gravitational field that couple to the mirrors in the detector.
 4. *Thermal noise* is caused by Brownian motion of the atoms and molecules in the suspensions, substrate and coating of the mirrors and by changes in optical properties of the mirrors due to temperature changes in the mirror coatings.
 5. *Technical noise* collects many further subdominant noise sources, including:
 - *angle and length control noise*, caused by control loops used for mitigating other more important noise sources and for locking the detector,
 - *laser-frequency noise*, caused by instability of the lasers frequency,
 - *Rayleigh scattering noise* due to residual gas in the imperfect vacuum system,
 - and *lines*, narrowband disturbances caused by resonance of the suspensions, the AC powerline and its harmonics and deliberately introduced lines for calibration purposes. Moreover, there exist many more instrumental lines often also of unknown cause. A catalogue of known lines is maintained by the LIGO collaboration in the Gravitational Wave Open Science Center (GWOSC)².

Space-based gravitational-wave detectors do not encounter seismic noise or gravity gradient noise but are not actively maintainable and have to fulfill special requirements for space travel,

²<https://www.gw-openscience.org/about/> [53]

3. Gravitational Waves

e.g. being light weight. This increases the technical noise, e.g. from thruster activity. Furthermore, the mirrors can no longer be considered close to each other breaking the approximation for eq. (3.28). Using the correct transfer function shows that the sensitive frequency range depends on the size of the detector [54]. Hence the sensitive frequency range for LISA is in the $\mathcal{O}(\text{mHz})$ regime and DECIGO is most sensitive around $\mathcal{O}(\text{dHz})$ [55] (cf. Fig. 3.4).

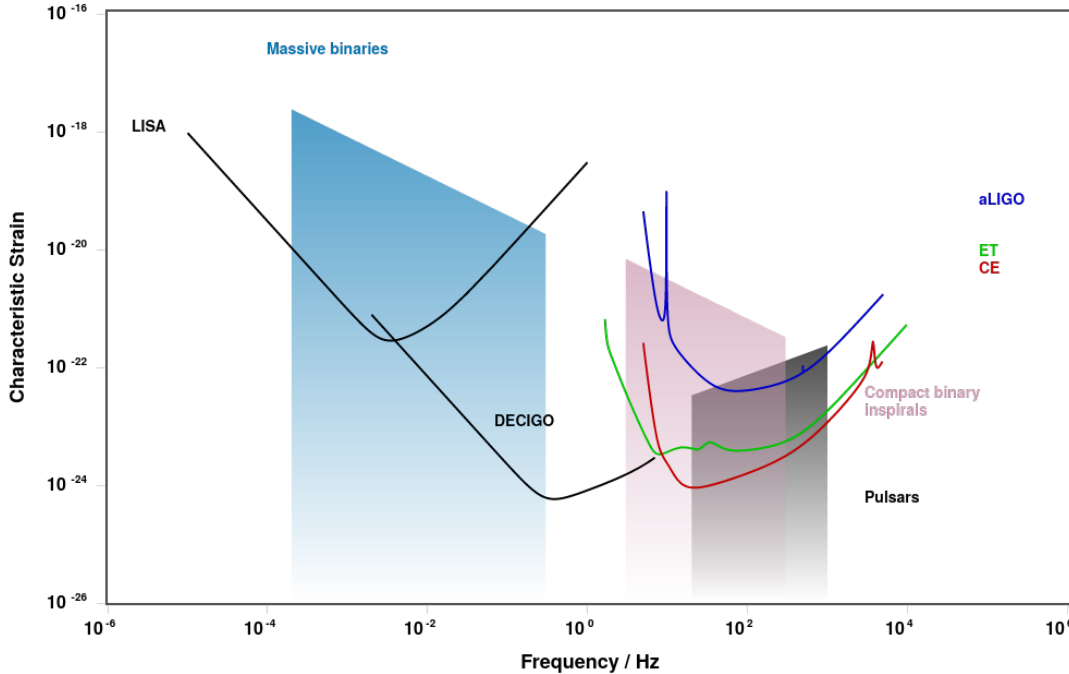


Figure 3.4.: Characteristic strain of the advanced LIGO detectors (aLIGO) at design sensitivity and for the planned detectors Einstein Telescope (ET), Cosmic Explorer (CE) as well as the planned space-missions LISA and DECIGO. The color trapezoid are an estimate of the characteristic strain of several sources (cf. sec. 3.4). This figure was created with the GW plotter web application (<http://www.gwplotter.com/>, which is based on [55]).

3.4. Astrophysical Sources of Gravitational Waves

Up to now we discussed how gravitational waves are a vacuum solution of the linearized Einstein field equations and how a detector has to be built to measure gravitational waves. In order to understand potential gravitational-wave signals, we have to look at potential sources.

In gravitational-wave data analysis the signals are typically divided in four categories according to duration and if the signal can be reasonably modeled. Short transient signals can either be from modeled sources, usually the merger of two compact objects, called *compact binary coalescence*, or from unmodeled sources, so-called *bursts* e.g. from supernova explosions. An overview of potential transient sources can be found in [56]. Long duration signals can either be individually modeled, *continuous gravitational waves* e.g. from non-axisymmetric rotating neutron stars, or an unmodeled *stochastic gravitational-wave background* from the early universe, mirroring the cosmic microwave background in the electromagnetic spectrum. While modeled searches usually can reach higher sensitivities for the specifically-modeled signals, unmodeled searches can potentially also detect gravitational waves from previously unknown emission mechanisms.

In this section we will briefly discuss signals from compact binary coalescences in the context of the first gravitational-wave observations and then concentrate on continuous gravitational waves throughout this thesis.

3.4.1. Compact Binary Coalescence

The only observed source of gravitational waves so far are mergers of compact objects, i.e. black holes and neutron stars. A signal from a merger of two black holes was first detected in 2015 [1], the first signal of two merging neutron stars was detected in 2017 [2]. Since then dozens of binary merger signals have been detected.

Two astrophysical compact objects orbiting each other radiate energy in the form of gravitational waves. The energy is drawn from the kinetic energy of the two objects. Therefore, the orbit of the binary is shrinking. As long as the two objects are far apart, their orbits can be considered to be Keplerian orbits without relativistic corrections [57]. This first phase is usually called the *inspiral* phase. In this phase the amplitude of the emitted gravitational wave is proportional to the inverse distance between the two objects (see [58]). Hence, with a large distance between the objects, the emitted gravitational wave is too weak to be detected. When the orbits are getting closer, the amplitude gets larger and the wave becomes detectable. On the other hand, for smaller orbits, we have to consider the first post-Newtonian relativistic corrections [59].

At some point the system has lost so much energy that the two objects touch each other. This is called the *merger* phase of the coalescence. In a short time around the merger the objects move at highly relativistic velocities and get tidally deformed so much that the post-Newtonian framework becomes insufficient [60]. At this point the signal can only be modeled by numerically solving the full Einstein equations [61, 62].

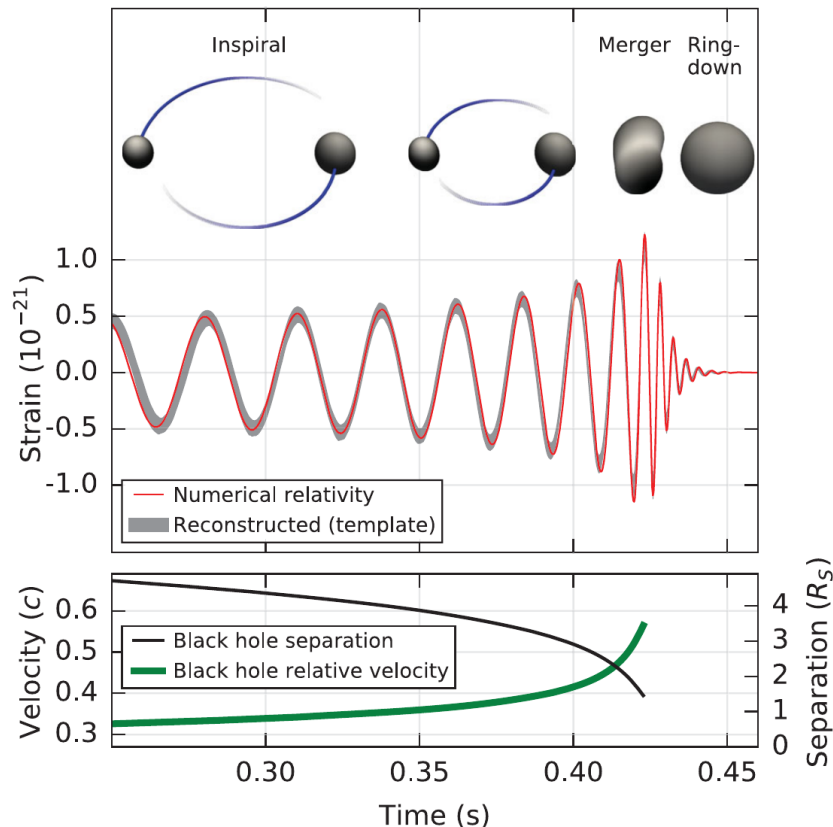


Figure 3.5.: GW150914: The first binary black hole merger observed by LIGO as velocity, separation and strain over time. The different phases of the coalescence, inspiral, merger and ringdown are illustrated by numerical relativity models of the black hole horizons at the top. The strain is reconstructed from the measured signal and compared to a numerical relativity calculation (Credit: Fig. 2 of [1]).

3. Gravitational Waves

The resulting object of the merger is initially highly deformed and emits gravitational radiation until it settles down either to a neutron star³ or to a black hole [63]. This is called the *ringdown* phase. The development of the coalescence signal is illustrated in Fig. 3.5. At the sensitivity of the current detectors the entire observable signal only lasts from fractions of a second for binary black hole signals up to tens of seconds for binary neutron star signals [2]. These short-lived transient signals have dimensionless strain amplitudes up to $\sim 10^{-21}$.

The observation of gravitational waves from binary mergers gives information about the prevalence of these objects in the universe and can for example be used to test general relativity [64, 65]. Binary-neutron-star merger signals can be accompanied by electromagnetic counterparts giving insight toward the neutron-star equation of state, i.e. the internal pressure and density structure of the neutron star [66].

3.4.2. Continuous Gravitational Waves

Continuous gravitational waves (CWs) are long-lasting signals with slowly varying intrinsic frequency. Several interesting potential sources are known, most notably rapidly rotating neutron stars with some kind of non-axisymmetry. Discovering a continuous gravitational wave could therefore improve our knowledge of these objects as well as allow for new tests of general relativity [67, 68].

Continuous gravitational waves have not been observed yet. Their expected amplitude is much lower than the amplitude of the coalescence signals discussed in Sec. 3.4.1. The best upper limits on the dimensionless strain amplitude from gravitational-wave-detector data are currently $\sim 2 \cdot 10^{-25}$ [69] for the entire sky and $\sim 10^{-26}$ for known pulsars. However, converting the observed change in frequency of known pulsars, the spindown \dot{f} , to a gravitational-wave amplitude with the assumption that all the energy loss is due to gravitational radiation often leads to lower upper limits. These *spindown upper limits* are as low as $\sim 10^{-27}$ for many known pulsars [70].

To compensate the lower amplitude CW data analysis requires the integration over long stretches of data. Therefore, CW searches are often limited in sensitivity due to the high computational cost of the analysis (cf. Ch. 4). Hence, improving the sensitivity of CW searches can be achieved by improving the detectors or the search methods.

Neutron Stars as Sources of Continuous Waves

Soon after the neutron was discovered by Chadwick [71], the possibility of an inverse beta decay of electrons and protons to neutrons at very high densities was considered and Baade and Zwicky [72] proposed a neutron star as a result of supernova explosions. In 1967 Pacini [73] postulated that a highly magnetized rotating neutron star would emit electromagnetic waves. These waves are emitted as a beam and therefore are only visible from Earth periodically if the beam points towards Earth. Neutron stars with these pulsating emissions are called *pulsars*. Almost simultaneously, graduate student Jocelyn Bell Burnell and her supervisor Antony Hewish already observed the first pulsar [74]. Since then over 1500 pulsars have been discovered.

According to its name, a neutron star is supposed to create the pressure necessary to resist the gravitational collapse by the quantum mechanical degeneracy pressure of neutrons from the Pauli exclusion principle. This simple degeneracy model can only support neutron stars with a maximum mass of 0.7 solar masses [75]. However, if we include repulsion of the neutrons due to the (strong) interaction, this mass limit, the Tolman-Oppenheimer-Volkoff limit, is increased dramatically. An accurate description therefore also has to model the thermodynamics and composition of nuclear matter [76]. A recent study considering multiple models of the neutron star equation of state estimated the maximum mass with the results from the neutron-star

³This is only possible for a merger without a black hole but with sufficiently-light neutron stars.

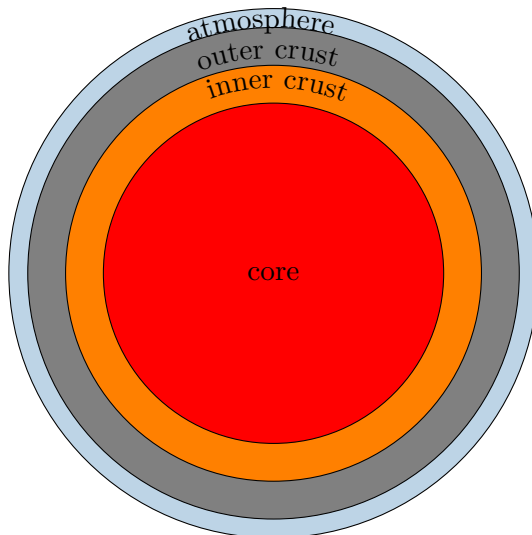


Figure 3.6.: Schematic of the structural layers of a neutron star: The neutron star is surrounded by a thin atmosphere of light gases. The outer crust consists of white-dwarf matter, heavy nuclei and electrons, the inner crust is made of nuclei, neutrons and electrons. The core composition is not well known but in the outer regions it likely consists of neutrons, protons, electrons and muons [78].

merger event GW170817 to be ~ 2.16 solar masses [66]. The most massive pulsar discovered is believed to have a mass of ~ 2.14 solar masses [77].

Nowadays the structure of a neutron star is believed to be roughly divisible into four distinct regions [78]:

1. The *core* with the highest pressure contains only neutrons, protons, electrons and muons but could also contain heavier baryons, and possibly free quarks. It is the largest part of the star with a radius of 9 – 12 km [79].
2. The *inner crust* contains nuclei, neutrons and electrons. It is expected to be 1 – 2 km thick [79].
3. The *outer crust* essentially mirrors white-dwarf matter: heavy nuclei and free electrons and is expected to be about 0.5 km thick [79].
4. The *atmosphere* of a neutron star is likely very thin and possibly contains hydrogen, helium and other light elements possibly up to carbon [80, 81]. It is only a few centimeters thick [78].

The pressure and density structure in a neutron star is described by the unknown neutron star equation of state. Many models for the equation of state exist and gravitational-wave observations can help with ruling out some of them [82].

The neutron-star crust or magnetic fields are believed to be able to support a “mountain” on a neutron star. Alternatively, such a mountain could also be maintained through accretion. In general such a mountain would be a non-axisymmetric distortion of the neutron star with an equatorial ellipticity

$$\varepsilon := \frac{|I_{xx} - I_{yy}|}{I_{zz}} \quad (3.31)$$

with the principle moments of inertia I_{jj} . The amplitude of the emitted continuous gravitational waves can be derived from the quadrupole formula (3.25) giving [83, 84]

$$h_0 = \frac{16\pi^2 G}{c^4} \frac{I_{zz} \nu^2}{d} \varepsilon \quad (3.32)$$

3. Gravitational Waves

with a frequency $f = 2\nu$, i.e. twice the rotational frequency ν [42]. This is often considered the most promising emission mechanism. However, even predictions based on this mechanism for gravitational-wave strain from known pulsars are usually several orders of magnitude lower than the current sensitivity of CW searches with advanced LIGO (see Fig. 3.7). Because it is assumed that most neutron stars in the galaxy are still unknown, there are also many searches for unknown neutron stars assuming that there might be some exceptional neutron stars with significantly stronger emission. These could then be detected even with current detectors. We will discuss CW searches and data analysis in more detail in Ch. 4.

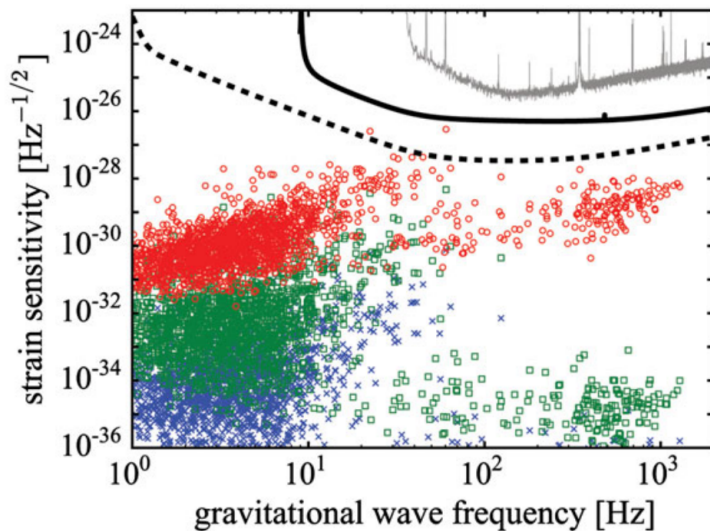


Figure 3.7.: Gravitational-wave strain predictions for known pulsars. The grey curve is the strain sensitivity during the S5 science run of initial LIGO assuming the analysis of two years of data. The black curve and the dashed black curve are the projected sensitivity for CW searches with advanced LIGO and with the planned Einstein Telescope, respectively. The differently colored points represent strain predictions from different models for neutron star matter and magnetic fields. For more details and the original figure see [85].

Another emission mechanism for CWs from neutron stars are non-axisymmetric instabilities. Especially r-modes, toroidal fluid oscillations driven by the Coriolis force of the rotating star are a promising emission mechanism [86, 87]. The r-mode instability could occur in newborn or rapidly-accreting neutron stars. The emission would occur at a frequency of approximately $f \approx \frac{4\nu}{3}$ [42] but a more accurate model is quite complicated [88].

The third major type of CW emission from a neutron star is free precession. It occurs if the neutron star's symmetry axis does not coincide with its spin-axis defining a wobble angle θ_W . The emission frequencies would be $f \approx \nu$ and $f \approx 2\nu$ [42].

The signal model presented in Ch. 4 is able to cover all these emission mechanisms because the main relevant difference is the relationship of gravitational-wave frequency to rotation frequency. If a CW search covers a large frequency band, a measured signal could originate from each of these emission models.

Further Continuous-Wave Sources

Another proposed potential source of CWs are *boson clouds* bound to black holes. These clouds would consist of electromagnetically invisible particles, i.e. dark matter. The idea is that bosons, such as axions, would be spontaneously created from the black holes rotational energy. Close to the horizon they could also encounter a boost in amplitude increasing the occupation number

[89] and increase the total mass. These bosons can then form a Bose-Einstein cloud, i.e. all the quanta would occupy only very few energy levels. These clouds can grow to significant masses and emit continuous gravitational waves due to annihilation or level transition [90, 91]. Multiple studies of the detectability of this type of continuous waves have been performed recently [89, 92–94].

Another promising source are binary systems of compact objects, mostly white-dwarf binaries but also binaries containing neutron stars or stellar mass black holes. Long before these objects merge, the emitted signal would be a continuous wave. However, the frequency of these waves would be far lower than the sensitive frequency range of the ground based detectors [95]. Therefore, observing the early evolution of binary systems and predicting their merger time is one of the main science goals of the planned space-based LISA detector [51] and studies for the data analysis of LISA data with respect to signals from white-dwarf binaries have been performed in [96] and [97].

4. Continuous-Wave Data Analysis

As this thesis is concerned with the sensitivity of searches for continuous gravitational waves and improvements to it, this chapter will give a brief introduction to continuous-wave data analysis.

We focus on the search for signals from isolated, rotating, non-axisymmetric neutron stars. However, as there is significant overlap to signals from neutron stars in binary systems, we will indicate at some points where a binary search would deviate.

4.1. Source and Solar System Barycenter Reference Frames

A gravitational wave in the source frame has only two polarizations (see Sec. 3.1). Therefore, we can write an arbitrary signal tensor $h^{ij}(\tau)$ as linear combination of the polarization basis tensors $(e_{\times}^{ij}, e_{+}^{ij})$ [40]:

$$h^{ij}(\tau) = h^{\times}(\tau)e_{\times}^{ij} + h^{+}(\tau)e_{+}^{ij} \quad (4.1)$$

where τ is the emission time of the signal at the source. The polarization basis can further be decomposed into a tensor product of vectors perpendicular to the direction of propagation. In this case the basis vectors \hat{l} and \hat{m} are usually fixed according to the source geometry, e.g. one of them as the projection of the spin axis of a rotating neutron star to the wave plane [40]. The wave plane is the plane perpendicular to the direction of propagation (see Fig. 4.1).

$$e_{\times}^{ij} = \hat{l}^i \hat{m}^j + \hat{m}^i \hat{l}^j \quad \text{and} \quad e_{+} = \hat{l}^i \hat{l}^j - \hat{m}^i \hat{m}^j \quad (4.2)$$

This basis is expressed in the source frame, but, in order to determine the influence of the gravitational wave at the detector, we usually want to switch to a polarization basis easily expressed in the Earth frame. To do that, we introduce basis tensors $(\varepsilon_{\times}^{ij}, \varepsilon_{+}^{ij})$ constructed from two vectors \hat{i}, \hat{j} defined with respect to the Earth (see Fig. 4.1).

$$\varepsilon_{\times}^{ij} = \hat{i}^i \hat{j}^j + \hat{j}^i \hat{i}^j \quad \text{and} \quad \varepsilon_{+}^{ij} = \hat{i}^i \hat{i}^j - \hat{j}^i \hat{j}^j \quad (4.3)$$

We fix one vector in the Earth's equatorial plane and the other, perpendicular to the first one, pointing to the Northern hemisphere, i.e. the projection of the Earth rotation axis onto the wave plane. The angle between the two polarization bases is called polarization angle ψ and is measured counterclockwise from \hat{j} to \hat{m} , i.e. $\cos \psi = \hat{j} \cdot \hat{m}$ (see Fig. 4.1). Note that this polarisation frame is approximately the same in the solar system barycenter frame (SSB)¹.

In order to fully convert equation (4.1) to strain in the detector, we need to perform two more steps [98]:

1. Convert the time of emission in the source frame to the time of arrival at the detector.
2. Consider the detector's response to signals from different sky positions dependent on its time-dependent orientation relative to the SSB.

The orbital motion of the Earth is known very accurately. Therefore, it is advantageous to split the transition from source to detector frame at the intermediate SSB frame. The distance

¹The motion of the Earth's axis due to precession and nutation is less than an arc-second per year. The stellar parallax due to the distance between SSB and detector is also much less than an arcsecond as typical CW sources are expected to be at least $\mathcal{O}(100)$ pc away.

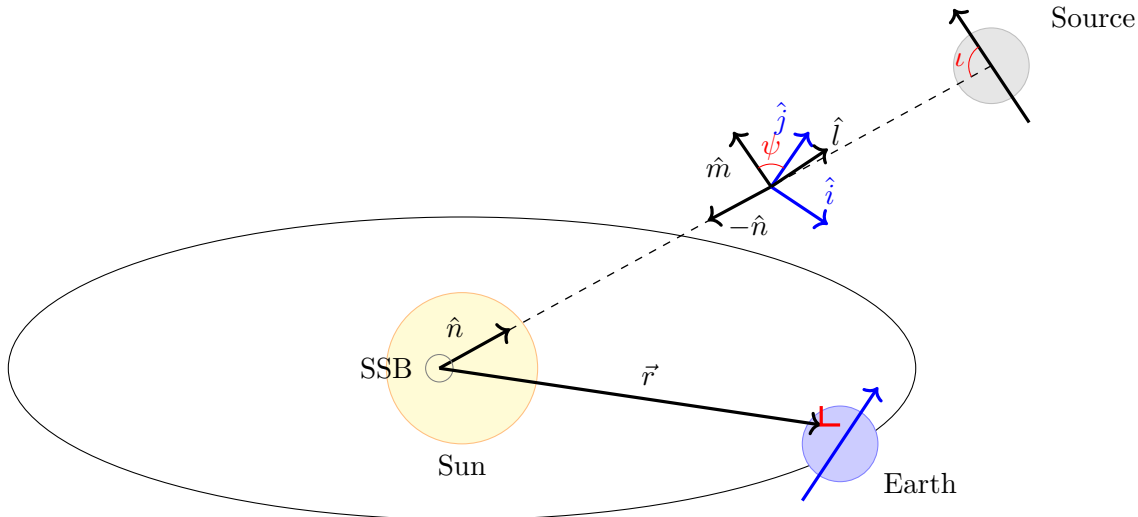


Figure 4.1.: Geometric relationship between SSB, detector and source for an incoming gravitational wave. The reference frames along the line of sight \hat{n} are the (\hat{l}, \hat{m}) frame usually aligned with source parameters and the blue (\hat{i}, \hat{j}) frame aligned with Earth rotation axis. The inclination angle of the source relative to the line of sight is denoted as ι .

of the source system barycenter to the SSB d is assumed to be constant². In the solar system we have to adjust for three different effects for the transition from detector frame to SSB frame [98]:

1. the Rømer delay, $\Delta_{R\odot}$, encoding the variations in the distance and hence light travel time to the pulsar due to orbital motion in the solar system,
2. the Einstein delay, $\Delta_{E\odot}$, the combined effect of gravitational redshift and time dilation due to the bodies in the the solar system,
3. and the Shapiro delay, $\Delta_{S\odot}$, caused by the wave propagating through the curved space time of the solar system.

In total, the detector time t is related to the source time τ as:

$$\tau = t + \Delta_{R\odot} + \Delta_{E\odot} - \Delta_{S\odot} - \Delta_{\text{source}} + \frac{d}{c} \quad (4.4)$$

where the typical convention to subtract $\Delta_{S\odot}$ is used. The wave travel time is written as the simple constant approximation $\frac{d}{c}$ and can be neglected [98]. Effects like microlensing could dynamically change the travel time and lead to deviations from the signal model presented here. A recent study regarding the effect of micro-lensing for CWs can be found in [99].

The source system correction term Δ_{source} is due to accelerated motion of the source. For example in a binary system the correction term can be expressed as

$$\Delta_{\text{source}} = \Delta_R + \Delta_E + \Delta_S \quad (4.5)$$

with Rømer, Einstein and Shapiro delay in the binary source system. More details on the timing relation in searches for signals from neutron stars in binary systems can be found in [100, 101]. For an isolated source this term vanishes: $\Delta_{\text{source}} = 0$.

²Only accelerated motions would break this assumption. Motion with a constant velocity only causes a constant Doppler-shift in frequency.

Expressions for the relativistic corrections can be found in [98]. Only for CWs with frequencies below 100 Hz the relativistic corrections are small enough to be neglected [102]. The Rømer delay Δ_R in the solar system can be expressed as

$$\Delta_{R\odot} = \frac{\vec{r}(t) \cdot \hat{n}}{c} \quad (4.6)$$

where $\vec{r}(t)$ is the time-dependent vector from the SSB to the detector (see Fig. 4.1).

For isolated sources the timing relation depends on the sky position \hat{n} of the source which can be expressed in equatorial coordinates as right ascension and declination (α, δ) . For sources from binary systems the timing relation also depends on the orbital parameters of the source system. In any case, it also depends on the detector location on the Earth, i.e. it must be calculated separately for each detector.

With the timing relation known, we can now write the source-frame equation (4.1) in the detector frame:

$$h^{ij}(t) = h_{\times}(\tau(t))e_{\times}^{ij} + h_{+}(\tau(t))e_{+}^{ij} \quad (4.7)$$

where we express the source polarization basis $(e_{+}^{ij}, e_{\times}^{ij})$ as rotation of the detector polarization basis $(\varepsilon_{+}^{ij}, \varepsilon_{\times}^{ij})$ by the polarization angle ψ (cf. eqs. (4.2), (4.2) and Fig. 4.1):

$$e_{+}^{ij} = \cos 2\psi \varepsilon_{+}^{ij} + \sin 2\psi \varepsilon_{\times}^{ij} \quad (4.8)$$

$$e_{\times}^{ij} = -\sin 2\psi \varepsilon_{+}^{ij} + \cos 2\psi \varepsilon_{\times}^{ij} \quad (4.9)$$

With the gravitational wave tensor h^{ij} at the detector determined, we can now calculate the detector response to the gravitational wave.

4.2. Detector Response

The detector response of an interferometric gravitational wave detector is given by relative distance changes between the interferometer arms. We make the reasonable³ assumption that the detector arm length L is much smaller than the wavelength λ_{GW} of the gravitational wave $L \ll \frac{\lambda_{\text{GW}}}{2\pi} = \frac{c}{2\pi f_{\text{GW}}}$. This effectively means that we can consider measurements as instantaneous because the metric h does not change during the travel time of the photon in the detector arms [39, 104]. This approximation allows us to write the scalar response $h(t)$ of a detector to an incoming gravitational wave with tensor $h^{ij}(\tau)$ as contraction of the detector tensor d_{ij} with the gravitational wave tensor h^{ij} :

$$h(t) = d_{ij}(t)h^{ij}(\tau) \quad (4.10)$$

where d_{ij} are the components of the detector tensor given by

$$d_{ij} = \frac{1}{2} (\hat{u}_i \hat{u}_j - \hat{v}_i \hat{v}_j) \quad (4.11)$$

with the unit vectors \hat{u} and \hat{v} pointing along the arms of the detector [96].

If we now use (4.7) to express (4.10) in the detector polarization basis we get:

$$h(t) = (h_{+}(\tau(t)) \cos 2\psi - h_{\times}(\tau(t)) \sin 2\psi) a(t; \hat{n}) + (h_{+}(\tau(t)) \sin 2\psi + h_{\times}(\tau(t)) \cos 2\psi) b(t; \hat{n}) \quad (4.12)$$

³This approximation breaks down when the wavelength is no longer small compared to the LIGO arm length of $4 \cdot 10^3$ m. A quick calculation shows that this occurs roughly at $\lambda_{\text{GW}} \gtrsim 10^5$ m, i.e. 3000 Hz. But usually CW searches only use the more sensitive range in the LIGO detectors of 20 – 1500 Hz (see Chapter 6). For larger interferometers a higher order approximation has to be used: the *rigid adiabatic(RA) approximation* [96, 103]

introducing the *antenna-pattern functions* as contraction [105]

$$a(t; \hat{n}) := d_{ij}(t) \varepsilon_+^{ij}(\hat{n}) \quad (4.13)$$

$$b(t; \hat{n}) := d_{ij}(t) \varepsilon_\times^{ij}(\hat{n}). \quad (4.14)$$

They encode the source location dependence of the detector response⁴.

4.3. CW Signal Model

Previously we only assumed that the gravitational-wave signal can be modeled as a plane wave, that the gravitational wave has a small amplitude such that the linear approximation of the Einstein equations holds and that the detector arm length is very small compared to the gravitational wavelength.

In this section we now want to specialize on the signal model for continuous gravitational waves (CWs). CWs are usually defined as long-lasting⁵ signals with slowly varying intrinsic frequency f (see Sec. 3.4.2). Thus we can model $h_{+,\times}(\tau)$ by

$$h_+(\tau) = A_+ \cos \Phi(\tau), \quad h_\times(\tau) = A_\times \sin \Phi(\tau) \quad (4.15)$$

with a series expansion of the Phase around an arbitrary reference time τ_{ref}

$$\Phi(\tau) = \phi_0 + 2\pi \sum_{s=0}^{N_{\text{SD}}} \frac{f^{(s)}(\tau_{\text{ref}})}{(s+1)!} (\tau - \tau_{\text{ref}})^{s+1} \quad (4.16)$$

with

$$\phi_0 := \Phi(\tau_{\text{ref}}) \quad (4.17)$$

$$f^{(s)}(\tau_{\text{ref}}) := \left. \frac{d^s f}{d\tau^s} \right|_{\tau_{\text{ref}}} = \left. \frac{1}{2\pi} \frac{d^{s+1} \Phi}{d\tau^{s+1}} \right|_{\tau_{\text{ref}}} \quad (4.18)$$

and the number of spin-down orders ($f^{(1)} = \dot{f}, f^{(2)} = \ddot{f}, \dots$) considered N_{SD} . As we already established in section 4.1 that the source time's τ relation to the detector time t depends on the sky position of the source \hat{n} , we determined all the *phase-evolution* parameters $\lambda = (\hat{n}, f, \dot{f}, \ddot{f}, \dots)$ affecting the phase evolution in our signal model. If we were considering a neutron star in a binary system, the orbital parameters of the source would also enter into the phase evolution as the timing relation would receive further corrections (see Sec. 4.1). The remaining parameters we established are called *amplitude* parameters and are denoted as $(A_+, A_\times, \phi_0, \psi)$.

If we now combine eq. (4.12) and (4.15) we can see that we can separate the two sets of parameters by introducing new quantities $h_\mu(t; \lambda)$ and $\mathcal{A}^\mu(A_+, A_\times, \phi_0, \psi)$:

$$h(t; \mathcal{A}, \lambda) = \mathcal{A}^\mu h_\mu(t; \lambda) \quad (4.19)$$

with an implicit sum over

$$\begin{aligned} \mathcal{A}^1 &= A_+ \cos \phi_0 \cos 2\psi - A_\times \sin \phi_0 \sin 2\psi, \\ \mathcal{A}^2 &= A_+ \cos \phi_0 \sin 2\psi + A_\times \sin \phi_0 \cos 2\psi, \\ \mathcal{A}^3 &= -A_+ \sin \phi_0 \cos 2\psi - A_\times \cos \phi_0 \sin 2\psi, \\ \mathcal{A}^4 &= -A_+ \sin \phi_0 \sin 2\psi + A_\times \cos \phi_0 \cos 2\psi, \end{aligned} \quad (4.20)$$

⁴Due to the Rømer delay given by Eq. (4.6) the final measured strain in the detector has an additional dependency on the sky position.

⁵Although the amplitude depends on the gravitational frequency (see eq. (4.23)) we assume constant amplitude in our signal model because the change in frequency is assumed to be small.

and

$$\begin{aligned} h_1(t; \lambda) &= a(t; \hat{n}) \cos(\Phi(\tau(t, \lambda)) - \phi_0), & h_2(t; \lambda) &= b(t; \hat{n}) \cos(\Phi(\tau(t, \lambda)) - \phi_0), \\ h_3(t; \lambda) &= a(t; \hat{n}) \sin(\Phi(\tau(t, \lambda)) - \phi_0), & h_4(t; \lambda) &= b(t; \hat{n}) \sin(\Phi(\tau(t, \lambda)) - \phi_0). \end{aligned} \quad (4.21)$$

This separation was coined *JKS factorization* after it was first introduced in the fundamental CW data analysis paper by Jaranowski, Krolak, and Schutz [102].

If the source of the CW is a non-axisymmetric neutron star (cf. Sec. 3.4.2), this expression can also be formulated with the overall gravitational-wave amplitude h_0 and the inclination angle ι of the neutron stars rotation axis relative to the line of sight (see Fig. 4.1). We replace

$$A_+ = \frac{1}{2} h_0 (1 + \cos^2 \iota), \quad A_\times = h_0 \cos \iota, \quad (4.22)$$

where the overall amplitude h_0 can be derived from the quadrupole formula [83, 84] to be:

$$h_0 = \frac{4\pi^2 G \varepsilon I_{zz} f^2}{c^4 d} \quad (4.23)$$

in terms of the equatorial ellipticity $\varepsilon := |I_{xx} - I_{yy}| / I_{zz}$, the moments of inertia I_{zz} along the spin-axis, I_{xx} and I_{yy} perpendicular to the spin axis and the distance d to the neutron star [106]. Therefore, the set of amplitude parameters is also often considered to be $(h_0, \cos \iota, \psi, \phi_0)$.

The signal model we constructed in this section is only true for a single detector. However, we can note that the amplitude parameters and therefore the \mathcal{A} 's actually do not depend on any detector-related quantity. Only the phase-evolution parameters are detector-dependent. This will be important in the next section where we look at how to analyze multi-detector data.

4.4. Matched-Filtering Searches

4.4.1. The Detection Problem: Frequentist and Bayesian approach

An interferometric gravitational-wave detector's output is contaminated by various noise sources as we discussed in section 3.3. Although we have now expressed the signal strain (cf. eq. (4.19)), we still need to distinguish that signal from the noise in the detector. The total detector strain data $x(t)$ can therefore be written as:

$$x(t) = n(t) + h(t; \mathcal{A}, \lambda) \quad (4.24)$$

From this equation onwards we mean by data x a vector of data from all detectors. If we need to explicitly show how multiple detectors are treated, we will introduce a detector index X : e.g. x^X .

At this point we now have to decide if we want to proceed with the *frequentist* interpretation of probabilities or the *Bayesian* interpretation. In this section we will briefly present both approaches.

Frequentist Approach

In the detection context we have to consider two hypotheses: the noise (or *null*) hypothesis $\mathcal{H}_N : h_0 = 0$ and the signal hypothesis $\mathcal{H}_S(\mathcal{A}, \lambda) : h_0 > 0$. Note that the null hypothesis is a special case of the signal hypothesis. For given data $x(t)$ we need to construct a *detection statistic* that tells us if we can reject the null hypothesis. According to the Neyman-Pearson-Lemma [107], the optimal decision between *simple* hypotheses is made by setting a decision threshold d_{th} on any monotone function $d(\Lambda(x))$ of the likelihood ratio $\Lambda(x)$ of the two hypotheses. The function $d(x)$ is then called the detection statistic. A simple hypothesis is a hypothesis that corresponds to a single fixed probability distribution. A *composite* hypothesis, on the other hand, does not fix

4. Continuous-Wave Data Analysis

the distribution completely but rather a family of distributions depending on a free parameter. For now let us assume that \mathcal{A} and λ are unknown but have fixed values, making our signal hypothesis a simple hypothesis. Then the likelihoods of the two hypotheses are given by

$$\mathcal{H}_S(\mathcal{A}, \lambda) : P(x|\mathcal{H}_S(\mathcal{A}, \lambda)) \quad (4.25)$$

$$\mathcal{H}_N : P(x|\mathcal{H}_N) \quad (4.26)$$

where the signal hypothesis likelihood depends on the parameters of the hypothetical signal. The likelihood ratio is therefore given by

$$\Lambda(x; \mathcal{A}, \lambda) = \frac{P(x|\mathcal{H}_S(\mathcal{A}, \lambda))}{P(x|\mathcal{H}_N)}. \quad (4.27)$$

The decision threshold d_{th} is usually determined for a fixed *false-alarm probability* p_{fa} :

$$p_{\text{fa}} := P(d(x) > d_{\text{th}}|\mathcal{H}_N) = \int_{d_{\text{th}}}^{\infty} P(d|h_0 = 0) dd, \quad (4.28)$$

the probability for pure noise data to produce a detection statistic value which exceeds the threshold. The probability for a signal in the data to be actually classified as such is called the *detection probability*:

$$p_{\text{det}} := P(d(x) > d_{\text{th}}|\mathcal{H}_S(\mathcal{A}, \lambda)) = \int_{d_{\text{th}}}^{\infty} P(d|\mathcal{A}, \lambda) dd \quad (4.29)$$

Equation (4.28) is the special case of $h_0 = 0$ of equation (4.29) because $\mathcal{H}_N = \mathcal{H}_S(h_0 = 0)$.

The sensitivity of CW searches is often given by the upper limits h_0^C . The frequentist upper limits give the weakest amplitude that can be detected with a probability C (typically either 90 % or 95 %), i.e. in n trials a signal with amplitude h_0 gives a detection statistic value exceeding the threshold in $C \cdot n$ of the n cases. This is usually determined by computationally expensive Monte-Carlo sampling to invert equation (4.29) for the amplitude h_0 . A sensitivity estimator as presented in Ch. 6 can give an estimate of the upper limits far more quickly. More details regarding the upper limit procedure are discussed in Sec. 6.4.

Until now we made the assumption that \mathcal{A} and λ are fixed and hence our signal hypothesis is simple. However, in a general CW search we assume that signals can have very different parameters \mathcal{A} and λ . This makes our signal hypothesis a composite hypothesis and therefore the Neyman-Pearson lemma is no longer applicable.

Dealing with composite hypothesis is therefore often done with the intuitive approach of the maximum-likelihood method, i.e. looking at the signal parameters which maximize the likelihood ratio [108]. The application of the maximum-likelihood method to the detection problem of continuous waves leads to the \mathcal{F} -statistic discussed in Sec. 4.4.3.

Bayesian Approach

The Bayesian version of hypothesis testing is called *model comparison*. While the frequentist approach assumed \mathcal{A} and λ to be unknown but fixed, in the Bayesian approach we always consider the values of \mathcal{A} and λ to be uncertain and take that into account by formulating a probability distribution for \mathcal{A} and λ before the experiment: the *prior probability*. We then use *Bayes' theorem*, a basic consequence of the definition of conditional probabilities,

$$P(\mathcal{H}|x, I) = \frac{P(x|\mathcal{H}, I)}{P(x|I)} P(\mathcal{H}|I) \quad (4.30)$$

to update the prior probabilities for any hypothesis \mathcal{H} with the information of a measurement $x(t)$ to retrieve *posterior* probabilities. In the context of the detection problem the prior probability

to consider is the probability $P(\mathcal{H}_S|I)$ to have a signal with any parameters (\mathcal{A}, λ) given the prior knowledge I . This information usually stems from previous measurements and from theoretical considerations but in principle can be chosen arbitrarily. The posterior probability $P(\mathcal{H}_S|x, I)$ is the probability for any signal with parameters (\mathcal{A}, λ) given the prior information I and the current measurement $x(t)$. They are related via the (marginal) likelihood $P(x|\mathcal{H}_S, I)$, which is the same as the likelihood of eq. (4.25) if the hypothesis is simple, i.e. if \mathcal{A} and λ are fixed. However, in this case we did not need to fix \mathcal{A} and λ and have a valid expression for composite hypothesis as well. The likelihood is calculated by marginalization over the parameters \mathcal{A} and λ :

$$P(x|\mathcal{H}_S, I) = \int P(x|\mathcal{H}_S(\mathcal{A}, \lambda), I)P(\mathcal{A}, \lambda|\mathcal{H}_S, I) d\mathcal{A} d\lambda. \quad (4.31)$$

The composite hypothesis \mathcal{H}_S and the simple hypothesis \mathcal{H}_N are exhaustive and mutually exclusive, i.e. either there is any signal or no signal. Therefore we can determine the denominator in (4.30), $P(x|I)$, from the normalization [108]

$$P(\mathcal{H}_S|x, I) + P(\mathcal{H}_N|x, I) = 1. \quad (4.32)$$

If we now want to compare the general signal hypothesis to the noise hypothesis we can construct the (marginal) likelihood ratio

$$B_{SN}(x|I) := \frac{P(x|\mathcal{H}_S, I)}{P(x|\mathcal{H}_N, I)} \quad (4.33)$$

which is called *Bayes factor*. This has been used to derive a Bayesian detection statistic [108] that is actually optimal in the Neyman-Pearson sense [109], if the priors match the actual parameter distributions, i.e. in general only in simulations with known distributions. The frequentist maximum-likelihood statistic for known phase-evolution parameters but unknown amplitude parameters has been rederived in a Bayesian way by assuming an unphysical uniform prior on the amplitude parameters [108, 110]. However, the frequentist maximum likelihood approach does require fewer integrations and is therefore usually computationally more efficient [108].

In the Bayesian approach model comparison is closely connected to *parameter estimation* as the posterior can be marginalized to give distributions of likely values for the separate parameters. Bayesian upper limits are determined by solving the equation

$$C = P(h_0 < h_0^C | x) = \int_0^{h_0^C} P(h_0 | x) dh_0, \quad (4.34)$$

stipulating that the real amplitude h_0 lies in the interval $[0, h_0^C]$ with confidence C . $P(h_0|x)$ is determined by marginalizing the other signal parameters, i.e. integrating $P(\mathcal{A}, \lambda|x)$ over all the other parameters. Thus, Bayesian upper limits have a very different interpretation than frequentist upper limits. We discuss this problem in more detail in Sec. 6.3.6. More information about the application of Bayesian statistics to CW searches can be found e.g. in [111, 112].

As most CW searches are working in the frequentist framework and only a few in the Bayesian context (cf. overview in Sec. 6.8), we will now focus on the frequentist approach for the remainder of this chapter.

4.4.2. Detector Data

In order to determine the likelihood of eq. (4.27) or a likelihood based detection statistic, we have to look at how to describe data in the detector.

In general the data in a detector might be non-complete and non-stationary. However, if we partition the data in short-duration discrete Fourier transforms called Short Fourier Transforms (SFTs) [113], we can assume stationarity over this shorter timespan. Using short timespans as basic data blocks makes it easier to exclude times with either very obvious large noise disturbances or times where the detectors did not produce science data. We will first look at the

4. Continuous-Wave Data Analysis

data properties in one SFT and later look at how to combine the data of multiple SFTs as first developed in [113–115].

When recorded in the detector, the data itself is sampled in discrete timesteps $t_j := j\Delta t$, giving samples $x_j^X := x^X(t_j)$. If we assume that the detector noises are uncorrelated, we can describe the noise samples $n_j^X := n^X(t_j)$ as draws from a multivariate Gaussian distribution $\mathcal{G}(0, \gamma_{ij}^X)$ with a covariance matrix $\gamma_{ij}^X = E[n_i^X n_j^X]$ for each detector. The single-sided power spectral density (PSD) in a SFT can then be estimated via the Wiener-Khintchine theorem as [42]

$$\mathcal{S}_{X\alpha}(f) \approx \frac{2}{T_{\text{SFT}}} E \left[\left| \tilde{n}_\alpha^X(f) \right|^2 \right] \quad (4.35)$$

with the Fourier-transformed noise series of one SFT

$$\tilde{n}_\alpha^X(f) = \Delta t \sum_j n_\alpha^X e^{-2\pi i f t_j} \quad (4.36)$$

with $n_\alpha^X(t) := n^X(t_\alpha + t)$ with t_α the start-time of the SFT α .

In the case of white Gaussian noise, which is sometimes used for testing purposes for example in the tests of the deep-learning methods presented in Chs. 7 and 8, this expression simplifies to

$$\mathcal{S}_{X\alpha} = 2 \frac{\sigma_X^2(\tilde{n}_\alpha^X)}{T_{\text{SFT}}} \quad (4.37)$$

given by the variance of the Fourier transformed data \tilde{n}_α .

The probability for data x to consist of draws from the Gaussian distribution with PSD \mathcal{S} can be given as [116]:

$$P(n|\mathcal{S}) = \kappa e^{-\frac{1}{2}(n|n)}. \quad (4.38)$$

The normalization factor κ will drop out in the likelihood ratio. The scalar product

$$(x|y) := \sum_{i,j} x_i (\gamma^{-1})^{ij} y_j. \quad (4.39)$$

is defined with the inverse of the covariance matrix γ^{-1} .

It has been shown [116–118] that this scalar product can be expressed as

$$(x|y) = 4 \sum_X \text{Re} \int_0^\infty \frac{\tilde{x}^X(f) \tilde{y}^{X*}(f)}{\mathcal{S}^X(f)} df \quad (4.40)$$

in the continuum limit $\Delta t \rightarrow 0$. This expression is a classical Wiener filter of matched-filtering theory [42] which is why many CW searches are usually referred to as matched-filtering searches.

We already established that we want to use the stationary Gaussian noise approximation only over the length of T_{SFT} . If we want to consider the long-lasting ($\gg T_{\text{SFT}}$) narrowband CW signals, we can express (4.40) semi-discretely in terms of per-SFT quantities:

$$(x|y) \approx 2 \sum_X \sum_{\alpha=1}^{N_{\text{SFT}}} \mathcal{S}_{X\alpha}^{-1}(f) \int_0^{T_{\text{SFT}}} dt x_\alpha^X(t) y_\alpha^X(t) \quad (4.41)$$

where $x_\alpha^X(t) := x^X(t_\alpha + t)$ with t_α the start-time of the SFT α and $\mathcal{S}_{X\alpha}$ the per SFT PSD according to eq. (4.35) [106].

We can now use (4.38) and (4.24) to find new expressions for the signal case likelihoods in (4.27):

$$P(x|\mathcal{S}, h_0 = 0) = P(n|\mathcal{S}) = \kappa e^{-\frac{1}{2}(n|n)} = \kappa e^{-\frac{1}{2}(x|x)} \quad (4.42)$$

$$P(x|\mathcal{S}, \mathcal{A}, \lambda) = P(x - h|\mathcal{S}, h_0 = 0) = \kappa e^{-\frac{1}{2}(x|x)} e^{(x|h) - \frac{1}{2}(h|h)} \quad (4.43)$$

Putting this back into (4.27) and taking the logarithm gives us a simple expression for the log-likelihood ratio:

$$\ln \Lambda(x; \mathcal{A}, \lambda) = (x|h) - \frac{1}{2}(h|h) \quad (4.44)$$

4.4.3. \mathcal{F} -statistic

The log-likelihood ratio can be used to gain insight whether a signal with parameters \mathcal{A}, λ is present in the data or not. In general, in a realistic CW search at least the Amplitude parameters \mathcal{A} , if not all the parameters, are unknown or only restricted to certain ranges. As mentioned in Sec. 4.4.1, we have to determine the maximum-likelihood parameters as an estimate of the real signal parameters and to decide if a signal is present or not.

In order to maximize the likelihood, we substitute (4.19) in (4.44) giving:

$$\ln \Lambda(x; \mathcal{A}, \lambda) = \mathcal{A}^\mu x_\mu - \frac{1}{2} \mathcal{A}^\mu M_{\mu\nu} \mathcal{A}^\nu \quad (4.45)$$

with definitions:

$$x_\mu(\lambda) := (x|h_\mu) \quad (4.46)$$

$$M_{\mu\nu}(\lambda) := (h_\mu|h_\nu) = \left(\frac{\partial h}{\partial \mathcal{A}^\mu} \middle| \frac{\partial h}{\partial \mathcal{A}^\nu} \right). \quad (4.47)$$

It is easy to show (first in [102]) that this expression can be maximized analytically for the amplitude parameters \mathcal{A} . We get as maximum likelihood amplitude parameters \mathcal{A}_{ML} :

$$\begin{aligned} \frac{\partial(\ln \Lambda)}{\partial \mathcal{A}^\mu} &= x_\mu - M_{\mu\nu} \mathcal{A}^\nu = 0 \\ x_\mu &= M_{\mu\nu} \mathcal{A}_{\text{ML}}^\nu \\ M^{\rho\mu} x_\mu &= M^{\rho\mu} M_{\mu\nu} \mathcal{A}_{\text{ML}}^\nu \\ M^{\rho\mu} x_\mu &= \mathcal{A}_{\text{ML}}^\rho. \end{aligned} \quad (4.48)$$

Here we introduced the inverse $M^{\rho\mu}$ of $M_{\mu\nu}$.

If we now resubstitute (4.48) in (4.45), we get a new detection statistic only dependent on the phase-evolution parameters λ :

$$2\mathcal{F}(x; \lambda) := 2 \max_{\mathcal{A}} \ln \Lambda(x; \mathcal{A}, \lambda) = x_\mu M^{\mu\nu} x_\nu \quad (4.49)$$

This statistic is called the \mathcal{F} -statistic of searches for continuous gravitational waves. It was first derived in [102] and then generalized for the multi-detector case in [117].

The $2\mathcal{F}$ -statistic follows a χ^2 -distribution with four degrees of freedom and non-centrality parameter $\rho^2 = (h|h)$, which is the square of the definition of the optimal⁶ *signal-to-noise ratio* (SNR) ρ , i.e. it is a central χ^2 -distribution in the pure noise case and a non-central χ^2 distribution in the signal case [102].

$$P(2\mathcal{F}|\rho^2) = \chi_4^2(2\mathcal{F}, \rho^2) \quad (4.50)$$

The expectation and variance of the $2\mathcal{F}$ -statistic are hence given by

$$E[2\mathcal{F}] = 4 + \rho^2 \quad \sigma^2[2\mathcal{F}] = 8 + 4\rho^2. \quad (4.51)$$

While the \mathcal{F} -statistic already maximizes the likelihood with respect to the amplitude parameters \mathcal{A} , to find the best signal parameters it must in general still be maximized with respect to the phase-evolution parameters λ . In the special case of a *targeted* search for CWs from known pulsars the phase-evolution parameters λ , frequency, spindowns, sky-position and possibly binary parameters are known and no further maximization is necessary. In all the other common search types *narrowband*, *directed* and *all-sky* searches a numerical search for the maximum likelihood parameters λ_{ML} is necessary. The differences between the search types are discussed in Sec. 6.1.

⁶Optimal in the sense that the matched-filtering for a signal h is performed with the exact model of the signal h . In contrast we can have a suboptimal SNR $\rho^2(h_S, h)$ where the template deviates from the signal $h = h_S + \delta h_S$ (see Sec. 4.4.4).

4.4.4. Templates and Computing Cost

In a wide-parameters space search the influence of a signal in parameter space is only present in a small area around the signal's parameters. Hence, many numerical maximization methods, e.g. Markov-Chain-Monte-Carlo (MCMC) methods, Newton method, etc. are unable to narrow in on the signal [119]. However, for example MCMC methods have been used for the smaller parameter space of follow-up searches [120, 121]. We will further discuss follow-ups in Sec. 4.4.5.

The numerical maximization over the phase-evolution parameters λ is therefore usually done by creating a template bank, i.e. a very dense grid in parameter space \mathbb{P} , and computing the \mathcal{F} -statistic for all these points. The sensitivity of such a search then depends on the resolution of the template bank. The loss of SNR due to lacking resolution is called template bank *mismatch*:

$$\mu(\lambda_S; \lambda) := \frac{\rho^2(\lambda_S; \lambda_S) - \rho^2(\lambda_S; \lambda)}{\rho^2(\lambda_S, \lambda_S)} \quad (4.52)$$

where

$$\rho^2(\lambda_S; \lambda) := E[2\mathcal{F}] - 4 \quad (4.53)$$

describes the SNR coming from a signal with parameters λ_S searched for with a template with parameters λ [118].

While the mismatch is not a symmetric quantity it can locally be approximated for small $\Delta\lambda = \lambda_S - \lambda$. The zeroth order trivially vanishes, while the first order term has to vanish in order for the \mathcal{F} -statistic to have a local extremum at the perfect match $\lambda = \lambda_S$ [118]. Thus, the lowest non-vanishing order is the second order and it can be used as approximate metric g_{ij} on the parameter space. Using a metric for the parameter-space manifold was first introduced in [122, 123] and later generalized for the multi-detector \mathcal{F} -statistic in [118].

$$\mu(\lambda_S; \lambda_S + d\lambda) = g_{ij}(\lambda_S) d\lambda^i d\lambda^j + \mathcal{O}(d\lambda^3). \quad (4.54)$$

Increasing the number of templates and therefore increasing template density will always improve the mismatch and hence the sensitivity of the search. However, the number of templates required to create a competitive search is very large. The computing cost typically grows linearly with the templates and thus gets so large that matched-filtering searches are limited in their sensitivity by computational cost. Therefore, finding a good approximation for the parameter space has been studied e.g. in [124].

It can be shown (e.g. in [42]) that the number of templates \mathcal{N} needed for a search with phase-evolution parameters $\lambda' = (\alpha, \delta, f, \dot{f})$, i.e. a search for an isolated neutron star where only the first order spindown is considered, scales approximately as

$$\mathcal{N} \sim T_{\text{span}}^5 f^2 d^4 \lambda', \quad (4.55)$$

i.e. with (at least) the fifth order of the observed timespan T_{span} .

The computing cost per template comes from an integration over the timespan of data, adding an additional order to overall computing cost $C_{\mathcal{F}}$, giving

$$C_{\mathcal{F}} \sim T_{\text{span}}^6 f^2 d^4 \lambda'. \quad (4.56)$$

On the other hand, the SNR of the signals is increased by increasing the amount of data used

$$\rho^2 \sim T_{\text{data}}, \quad (4.57)$$

where T_{data} is the total amount of data of all detectors [102]. T_{data} is not to be confused with the observation time T_{span} , the timespan from the first data point to the last data point. Thus, we want to make T_{data} (and not T_{span}) as large as possible given our fixed computing-cost budget. The only ways around the computing-cost trade-off are including data from more

equally sensitive detectors in the search, reducing the amount of data gaps and eliminating noise sources in the detector. However, this is usually not a question of data analysis but a question of detector construction and operation and therefore illustrates the necessity for close cooperation between experimentalists and data analysts.

For coherent matched-filtering searches this means $T_{\text{span}} = \mathcal{O}(\text{days}) - \mathcal{O}(\text{weeks})$ with the maximal currently available computing power, e.g. $\lesssim 44$ days in a recent directed search for CWs from supernova remnants [125]. This is much less than the expected lifetime of a CW signal and also much less than the typical duration of LIGO science runs of $\mathcal{O}(\text{months}) - \mathcal{O}(\text{years})$. Accordingly, for wide-parameter space searches it is possible to find more sensitive methods than coherent matched-filtering for a fixed computing cost. The most common approaches are semi-coherent matched-filtering searches.

4.4.5. Semi-coherent Searches and Follow-ups

Methods for wide parameter searches try to minimize the loss in sensitivity while reducing the computing cost as much as possible, i.e. they try to optimize the sensitivity at fixed computing cost.

Semi-coherent methods [126] reduce the coherent matched-filtering computing cost by reducing the coherently analyzed timespan $T_{\text{seg}} < T_{\text{span}}$ and combine multiple segments N_{seg} incoherently to a detection statistic over T_{span} . These methods are called *semi-coherent* search methods. In the easiest case the segments consist directly of the SFTs calculated from the raw detector data, i.e. typically $T_{\text{SFT}} = 1800$ s (used e.g. in [127]). However, if more computational resources are available, for example in Einstein@Home searches, longer segments are used to achieve better overall sensitivity, e.g. up to $T_{\text{seg}} \approx 20$ days in the most recent Einstein@Home⁷ search [128].

A semi-coherent search therefore always consist of at least two steps. First calculating a detection statistic over short segments while maintaining phase coherence and secondly combining these segments. In this section we want to focus on \mathcal{F} -statistic-based semi-coherent searches, especially the *StackSlide* [129] and the Hough- \mathcal{F} [130] methods. A list of CW searches performed with these methods as well as with other semi-coherent methods can be found in 6.8. However, most methods can be identified as implementations of the StackSlide method or variants to the Hough- \mathcal{F} method with different detection statistics and implementations. Further notable semi-coherent methods are the PowerFlux or Falcon methods [131–133], cross-correlation methods [134, 135], the TwoSpect algorithm [100] and Viterbi methods [136, 137].

We calculate the final detection statistics $\hat{\mathcal{F}}$ for StackSlide and n_c for HoughF as

$$\hat{\mathcal{F}} := \sum_{l=1}^{N_{\text{seg}}} \mathcal{F}_l \qquad n_c := \sum_{l=1}^{N_{\text{seg}}} \Theta(\mathcal{F}_l - \mathcal{F}_{\text{th}}) \qquad (4.58)$$

where \mathcal{F}_l is the coherent \mathcal{F} -statistic in segment l , N_{seg} is the number of segments and \mathcal{F}_{th} is a common per-segment threshold on the coherent \mathcal{F} -statistic for the Hough- \mathcal{F} method. $\Theta(x)$ is the Heaviside step function. The StackSlide detection statistic $\hat{\mathcal{F}}$ is usually referred to as semi-coherent \mathcal{F} -statistic and the Hough statistic n_c as number count, as it counts the number of threshold crossings. Other Hough variants usually only deviate from Hough- \mathcal{F} by using a different coherent statistic. PowerFlux and its successor Falcon are derived from the StackSlide methods, using weights to reduce the influence of disturbances and improving the handling of the direction-dependent detector antenna patterns [42].

The template grid needed to create a sensitive semi-coherent search is still very fine. However, it turns out that often fewer templates are needed to calculate the coherent detection statistic values than for the incoherent statistic of eq. (4.58) (see e.g. [138]) leading to two different

⁷Einstein@Home is a volunteer-computing project currently achieving a computational power of ~ 7 PFLOPS [7].

4. Continuous-Wave Data Analysis

template grids: the coarse grid for the coherent step and a fine grid for the incoherent step which usually uses a nearest-neighbor interpolation on the coherent statistic values to calculate (4.58).

Using these semi-coherent methods reduces the templates needed for a parameter space region $d^4\lambda'$ to

$$\hat{\mathcal{N}} \sim T_{\text{span}} T_{\text{seg}}^4 f^2 d^4\lambda' \quad (4.59)$$

giving a total computing cost scaling of

$$C_{\hat{\mathcal{F}}} \sim T_{\text{span}}^2 T_{\text{seg}}^4 f^2 d^4\lambda' \quad (4.60)$$

effectively replacing T_{span}^4 with the much smaller T_{seg}^4 [42].

Another method to improve sensitivity while avoiding large increases in computing cost are *Follow-up* searches or *Hierarchical* searches [126, 139]. These are done by first executing a search with a relatively high false-alarm level, i.e. low threshold, often called stage-0, and then applying additional methods to gradually eliminate the false alarms. The further stages are usually designed such that they have a very low false-dismissal probability $\gtrsim 99\%$ such that as many potential signals as possible get to the next stage. In each successive stage the parameter space is shrunk to an area around the candidates from the previous stage, the length of the coherent segments and the detection threshold are increased. While noise candidates are in general expected to either decrease or remain the same in detection statistic over the stages, signal induced candidates should increase in detection statistic value and hence also pass the next thresholds (see e.g. [128]).

As the number of candidates decreases from stage to stage, the higher template density and longer coherence times remain affordable. In fact, the total computing cost is usually dominated by stage-0. However, as mentioned in Sec. 3.3, there are also non-Gaussian disturbances in the data. For CW searches mainly lines, narrowband spectral artifacts, e.g. at the electric grid frequency, cause problems as they can appear quite similar to real signals and therefore can survive the multiple stages.

To reduce the number of candidates caused by lines or other disturbances, many searches use additional procedures, e.g. removing line disturbances from the detector data before the stage-0 search [140], applying veto methods (e.g. [141]) and constructing alternative, more robust detection statistics [142, 143]. Another approach is to cluster candidates with very similar parameters together and treat them as a single candidate [18, 144]. Moreover, completely different search methods for the follow-up have also been tested [17, 121].

In addition it is part of ongoing research to find computationally more efficient implementations of semi-coherent searches [138, 145]. However, this limitation by computational cost motivates the search for alternative search methods such as a deep-learning search for CWs as presented in Chs. 7 and 8 of this thesis.

5. Deep Learning

5.1. Deep Learning and Deep Neural Networks

Deep Learning is a specific kind of machine learning, which is a part of the research on artificial intelligence. I want to briefly introduce the concepts of artificial intelligence and machine learning and then focus on deep-learning and deep neural networks.

Defining artificial intelligence (AI) is difficult as not even intelligence is well defined but rather known as an intuitive concept. Intelligence is usually associated with making good and foresighted decisions in the context of your environment. According to that, animals and humans possess different levels of intelligence. Artificial intelligence is the task of creating a machine or machine-run algorithms that can perform tasks and solve problems usually requiring human or animal level intelligence [146, 147].

The idea of artificial intelligence is very old. Even in antiquity Aristotle dreamed of automated tools, which could replace servants and slaves [146]. With the development of computers this idea seemed suddenly to be in reach of technology causing a first wave of AI research in the 1950s. The first AI programs of this era were implemented as *symbolic* artificial intelligence: A program with a fixed set of rules such that it can solve the specific task it was designed for. A typical example for symbolic AI are so-called *expert systems*, programs that supposedly could replace a human expert in a field. For example systems that are predicting pre-term birth risks [148] or monitoring the Space Shuttle mission control [149].

In the 1980s a new approach was developed: machine-learning [147]. Instead of fixing the rules and applying them to data, the task for the programs was changed to finding the necessary rules from the given data itself. The learned set of rules can then be applied to a new data set. We can also imagine the training of a machine-learning algorithm as finding a series of transformations of the data to an optimal representation. The allowed transformations are usually determined before the learning process and can be coordinate changes, linear projections, translations or nonlinear operations. In Fig. 5.1 we show as an example how finding the right transformation can make a classification of data points easier.

Since its invention, machine-learning has been implemented with many different algorithms and methods. They can be classified according to three different properties [150]:

- Either an algorithm is trained with human supervision, i.e. the algorithm is given the desired output of training examples, *supervised* learning, or the algorithm has to learn without labels, *unsupervised* learning.
- Either the algorithm learns from fixed size data sets, *Batch* learning, or from a continuous data stream, *online* learning.
- Either the algorithm compares a new instance of data to a set of labeled training instances and chooses the new label from the most similar example, *instance-based* learning, or the algorithm infers a model from the training data which is then used for example to classify new data, *model-based* learning. For example, a spam-filter that looks for specific words occurring in previous spam mails to flag a new mail as spam would be instance-based, while fitting a set of data points with a function would be an example of model-based learning.

For continuous-gravitational-wave (CW) searches the training data has to be created with artificially injected signals because training needs thousands of signals and none have been

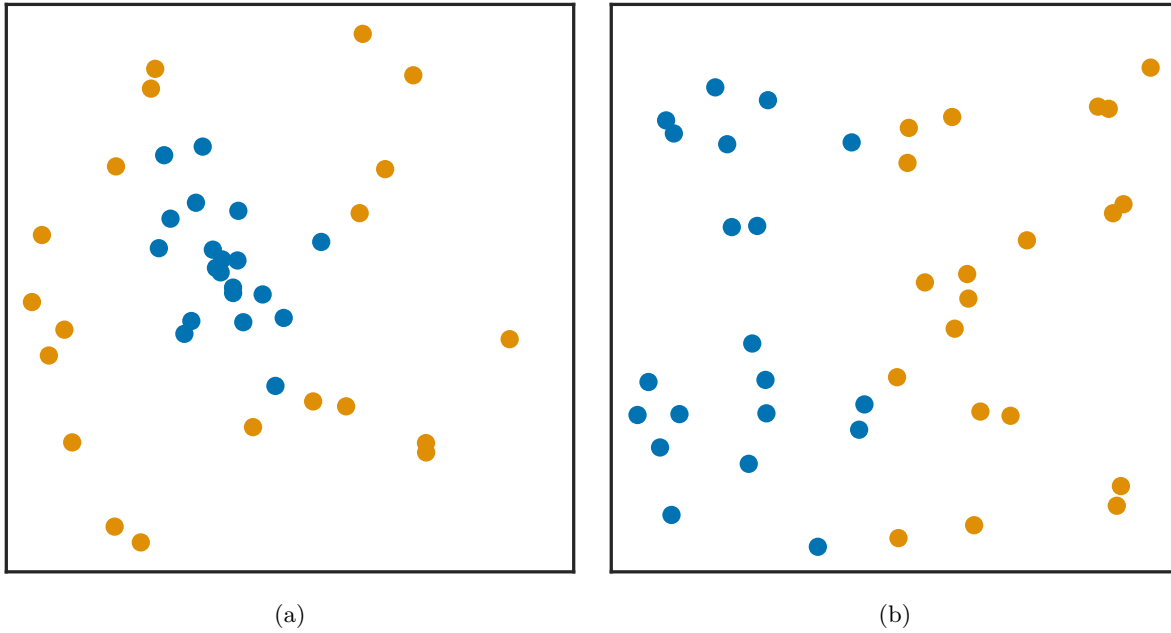


Figure 5.1.: Machine-learning algorithms find transformations to represent data more simply. In the coordinate system (Cartesian) in (a) the points are clearly not separable along one variable. If we apply a transformation to polar coordinates (b), we can separate the points along one variable.

detected yet. Therefore, labeling the data with the desired output is easy and the typical downside of supervised learning - the time-consuming step of labeling data - is moot. Thus, using a supervised algorithm is sensible, as they are usually more capable than unsupervised algorithms, if the labels are available. Due to the size of the searched parameter space (cf. Sec. 4.4), it is likely that an instance-based approach would require large amounts of computational resources, because the complexity of instance-based learning is growing with the amount of data. Therefore, a model-based approach seems more promising.

While it is possible to generate a continuous stream of CW signals in Gaussian noise with the existing tools of the LALSUITE software package, for the work in Chs. 7 and 8 we decided to make use of pre-generated fixed data sets. This was easier to implement efficiently with the deep-learning toolkit KERAS [151] than a data stream because a data stream always has to generate new data faster than the network can process it independent of the used hardware.

As discussed above, machine-learning algorithms are trained to find an optimal data representation by learning to apply transformations. The fundamental idea of deep-learning is to learn a complex transformation to optimize the final representation by decomposing it into a sequence of simple transformations. This is done in the form of *layers*. Each layer implements a transformation from one data representation to the next one. Typically, this layered approach is implemented as a neural network: if we stack more than two layers, the network is called a *deep neural network*; a network with one or two layers is sometimes referred to as *shallow neural network* [147].

In the context of the CW search method developed in this thesis (see Chs. 7 and 8), we decided to use a deep-neural network trained as a noise-versus-signal classifier. *Deep neural networks (DNNs)*, a model-based approach, have recently shown remarkable capabilities: from image and speech recognition to language translation, DNNs dominated recent machine-learning competitions [8]. Furthermore, DNNs have also been shown to be useful tools in many different aspects of gravitational-wave research [9–27].

As the naming suggests, the smallest structural unit of a neural network is the *neuron*. Its

working principle was designed after neurons in human or animal brains, which, dependent on a combination of the strength of their input connections, either do output a signal (“fire”) or not. However, this original idea of modeling the way human or animal brains work, is nowadays mainly reflected in vocabulary. Neurons in a modern neural network behave significantly differently than neurons in the human brain [147].

5.2. Neurons, Layers, Networks

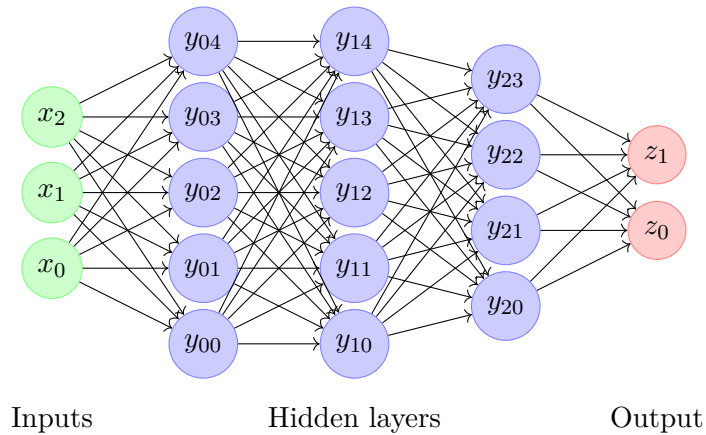


Figure 5.2.: A dense neural network with three inputs (green), two outputs (red) and three hidden layers (blue).

To understand the working principle of neural networks, we want to look at neurons in the simplest layer, the *dense* layer. In a dense layer every neuron is connected to every neuron of the previous layer (see Fig. 5.2).

The i -th neuron in a dense layer can be modeled as functions $f_i : \mathbb{R}^n \times \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ of two vectors: the inputs to the neuron \vec{x} as well as the intrinsic weights and the bias of the neuron (\vec{w}_i, b_i) :

$$f_i(x; \vec{w}_i, b_i) := a(z_i) := a\left(\sum_{j=1}^n w_{ij}x_j + b_i\right) \quad (5.1)$$

where $a : \mathbb{R} \rightarrow \mathbb{R}$ is called activation function and z is the pre-activation output (illustrated in Fig. 5.3) [152].

The x_j are the inputs to the neuron, i.e. either the input to the network or the output of neurons of the previous layer. In a dense layer the number of inputs and weights per neuron is

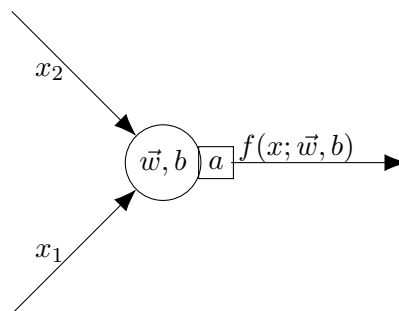


Figure 5.3.: A single neuron of a neural network. In this example the neuron has two inputs, two weights \vec{w} , a bias b and an activation function a .

5. Deep Learning

always equal. We therefore have one weight for every input and the bias value. However, we will see later that it can be beneficial to reuse weights for several inputs (see Sec. 5.5.1).

The bias b_i acts as a threshold at the point of the weighted sum of inputs, where the behavior of the activation function changes. To ease notation, from now on we will consider the bias as another weight $w_{i0} := b_i$ which always has an input of $x_0 := 1$, giving the neuron equation as

$$f_i(x; w_i) = a\left(\sum_{j=0}^n w_{ij}x_j\right). \quad (5.2)$$

Note that from now on we will implicitly include the bias if we mention the weights of the DNN.

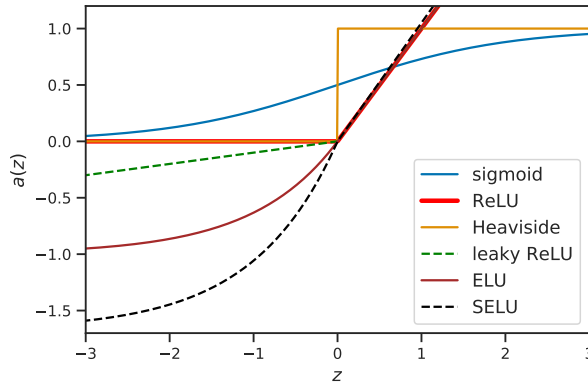


Figure 5.4.: Six important activation functions in neural networks: Sigmoid, rectified linear unit (ReLU), Heaviside step function, leaky ReLU, exponential linear unit (ELU) and scaled ELU (SELU).

The weights w_i in eq. (5.2) are the trainable parameters of a neuron. Therefore, each neuron possesses $N_{\text{weights}} := n + 1$ trainable parameters, where N_{weights} is determined by the number of connections n the neuron has to other neurons.

The activation function a is the same for all neurons in a layer, often it is even the same for all but the output neurons in a network. We consider the activation function as part of the neuron and its layer, but sometimes it is also considered to be a separate layer. The activation function must be at least piecewise differentiable with respect to all the weights $w_i \in \mathbb{R}$. If the activation function is a non-linear function, then it can be shown that a neural network can approximate every function if the DNN has enough neurons. This is called the *universal approximation theorem* (see [153–155]).

The step function

$$\Theta(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases} \quad (5.3)$$

is one of the simplest activation functions. A neuron using it is sometimes also called a perceptron. It was invented in 1957 by Frank Rosenblatt [150]. Using only a multi-layer perceptron network, it can be shown that all the basic logic gates can be built by setting the weights of a multi-layer perceptron ([155], see e.g. [150] for the “Xor”-gate).

However, over time, it turned out that other activation functions lead to more success in training neural networks. Rumelhart, Hinton, and Williams [156] introduced the sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (5.4)$$

and more recently the rectified linear unit (ReLU)

$$r(z) = \max(0, z) \quad (5.5)$$

brought further improvements for training a neural network [157, 158]. Nowadays, many different activation functions are used. Some popular examples illustrated in Fig. 5.4 are leaky ReLU [159]

$$l(z) = \max(0.01z, z), \quad (5.6)$$

exponential linear unit (ELU) [160] and scaled ELU (SELU)[161]

$$s(\alpha, z, \lambda) = \lambda \begin{cases} \alpha(e^x - 1) & x < 0 \\ x & x > 0 \end{cases} \quad (5.7)$$

with $\lambda = 1$ and arbitrary α for ELU and $\lambda = 1.0507$ and $\alpha = 1.67326$ for SELU.

The activation function in the output layer usually deviates from the activation function of the remaining layers and is chosen such that error calculations (cf. Sec. 5.3) can be done more easily. In classifiers the activation function of the last layer is usually a softmax function

$$s_i(x) = \frac{e^{x_i}}{\sum_{j=1}^{N_{\text{outputs}}} e^{x_j}}, \quad i = 1, \dots, N_{\text{outputs}} \quad (5.8)$$

making sure the output values are adding to one and therefore can be interpreted as probability estimates for the respective classes [150].

In a DNN neurons are arranged in layers (see Fig. 5.2). The layers are connected by the functions of the neurons as given in eq. (5.2). All neurons of a layer combined form a transformation from one data representation to the next one. Depending on the weights of the neurons the layers perform different transformations.

A neural network consists of three types of layers, the *input layer*, the *output layer* and the layers in between, called *hidden layers*. The output layer or simply output must have the size of the desired answer. The size is the number of classes in a classification task and e.g. the same size as the input for a neural network trained to manipulate images. The hidden layers are the ones actually performing transformations to get from the input representation of the data to the output representation. Geometrically these transformations can be imagined as trying to unfold a crumpled paper ball back to a flat sheet of paper. Like humans would try to unfold the paper step by step, each hidden layer in the network slightly unravels a highly folded manifold [147].

The input layer or simply the input is the data that the network is supposed to analyze. For the input size we typically distinguish two types of dimensions: the *input dimensions* and the *channels dimension*. For example, a RGB image would be an input with two input dimensions, height and width, and a third dimension containing the three *channels* that encode the red, green and blue part of the picture. A single instance of data, e.g. a single image, given to the network is often referred to as (input) *sample*. Training a network typically requires many data samples. The entirety of samples used for training is called the *training set*.

5.3. The Training Loop of a Neural Network

5.3.1. Loss and Gradient Descent

We have already established that each layer in a network possesses trainable parameters called weights and that a machine-learning algorithm is supposed to learn a set of rules from a given labeled training set. This translates to finding the best possible weights such that the input is transformed with a sequence of simple transformations to give back the label of the respective data sample. But how does it perform this task?

For the network to be able to improve itself, it has to have some kind of measure on how wrong a *prediction* – the result of applying the network to a data sample \mathbf{x} – is. For that purpose we define a *loss* (or *cost*) function $\mathcal{L} : \mathbb{R}^{N_{\text{out}}} \rightarrow \mathbb{R}$ that measures the distance between the N_{out} network predictions \mathbf{y}^N and *labels* \mathbf{Y} associated with this data sample. In classification

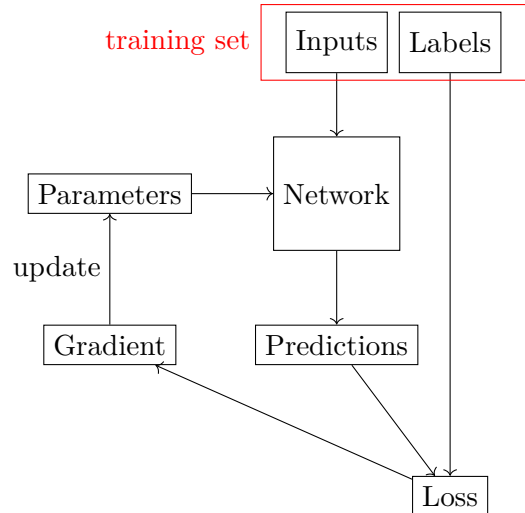


Figure 5.5.: The training loop of a neural network consists of the forward pass and the backward pass. In the forward pass the network is evaluated for the inputs from the training set. The output is a prediction of the label of this input sample. The predictions are then compared to the labels of the training set to calculate the loss. In the backward pass the gradient of the loss with respect to the weights is calculated and then used to update the weights of the network via *gradient descent*.

the number of labels equals the number of classes. The label Y_k for class k is 1 if the sample belongs to that class, otherwise it is zero. The loss function is then used as feedback to adjust the weights to get a result closer to the label [147]. In this section we count the layers with an upper index and components with lower indices. The network output is the output of the last (N -th) layer, hence the notation \mathbf{y}^N .

A typical loss function for a regression task, e.g. predicting the SNR of continuous gravitational waves, is the *mean-squared-error* function:

$$\mathcal{L}(\mathbf{y}^N) = \frac{1}{N_{\text{out}}} \sum_{k=1}^{N_{\text{out}}} (y_k^N - Y_k)^2. \quad (5.9)$$

For classification, as used in Chs. 7 and 8, the most common loss function is the categorical cross-entropy

$$\mathcal{L}(\mathbf{y}^N) = - \sum_{k=1}^{N_{\text{out}}} Y_k \ln y_k^N \quad (5.10)$$

$$N_{\text{out}}=2 \quad -Y_1 \ln y_1^N - (1 - Y_1) \ln(1 - y_1^N),$$

which is also called binary cross-entropy in the $N_{\text{out}} = 2$ case [150]. The loss function has to be chosen such that minimizing it correlates with success for the task at hand. It is not always possible to optimize directly for the metric that measures success on a problem because the loss must be differentiable and ideally calculable for a single instance of data [147]. Furthermore, the loss calculation has to be fast in order to prevent slowing down the training. For example, the detection probability at a fixed false alarm level is not suited as a loss, because we would need many noise samples without a signal to determine a false alarm threshold and then many signal samples to calculate the detection probability (cf. Sec. 7.2.1). This calculation can only sensibly be done for the entire training set and is much too slow to perform during the training of a neural network. However, the detection probability or other quantities can be calculated additionally to measure the performance of a neural network and ensure that reducing the loss correlates with an improved performance.

Theoretically, the easiest way to get to parameters with smaller loss is to determine the (negative) gradient of the loss with respect to the weights and follow it towards the global minimum¹ by alternately updating the parameters and recalculating the gradient. This method is called *gradient descent* and is an essential part of the core loop of the neural network training. A network is usually initialized with random parameters, i.e. random data transformations, and then the training loop is repeated for many thousands of training samples until the loss is sufficiently minimized [147].

Determining the gradient in an efficient way is usually done with the *backpropagation* algorithm. It was first introduced in the 1970s but became popular after Rumelhart, Hinton, and Williams [156] discovered that it made learning of neural networks significantly faster than previously used methods [152].

The goal of the backpropagation algorithm is to calculate the gradient, i.e. the partial derivatives $\frac{\partial \mathcal{L}}{\partial w_{ij}^l}$ of the loss function \mathcal{L} with respect to the weights w_{ij}^l . Our notation here symbolizes that the weight w_{ij}^l is the weight of the connection of the j -th neuron in the previous layer to the i -th neuron of the layer l . If the network contains M weights calculating the finite differences $\frac{\mathcal{L}(w_{ij}^l) - \mathcal{L}(w_{ij}^l - \delta w_{ij}^l)}{\delta w_{ij}^l}$ for all weights w_{ij}^l in all layers l would require $M + 1$ passes through the entire network. As the network can contain a long chain of transformations and thousands to millions of parameters, it is therefore usually prohibitive to calculate the finite difference for every parameter. Instead backpropagation uses the chain rule of differentiation to pass through the network backwards avoiding duplicate calculations.

Let us consider a network with N layers. We call the input (\mathbf{x}, \mathbf{Y}) , where \mathbf{x} is the data sample and \mathbf{Y} its label. Each layer l performs a transformation $T^l = a^l \circ \mathbf{w}^l$ consisting of the activation function a^l and a matrix product of the weights \mathbf{w}^l with the input \mathbf{x} . The output of layer l is therefore generalized from (5.2) as

$$y_i^l = (T^l(\mathbf{y}^{l-1}))_i = a^l(z_i^l) = a^l \left(\sum_j w_{ij}^l y_j^{l-1} \right) \quad (5.11)$$

where \mathbf{z}^l denotes the pre-activation output of layer l . With this notation we can now look at how an input passes forward through the network during evaluation and how it passes backwards during gradient calculations:

1. **Forward pass:** The input \mathbf{x} is given to the network and passed to the neurons in the first hidden layer with weights \mathbf{w}^1 . The first hidden layer applies a transformation T^1 giving

$$y_i^1 = (T^1(x))_i = a^1 \left(\sum_j w_{ij}^1 x_j \right) \quad (5.12)$$

and passes the result to the second layer, which applies T^2 giving $\mathbf{y}^2 = T^2(\mathbf{y}^1)$. This repeats until we reach the output layer N with output $\mathbf{y}^N = T^N \circ T^{N-1} \circ \dots \circ T^1(\mathbf{x})$. Then the loss $\mathcal{L} = \mathcal{L}(\mathbf{y}^N(x, w), \mathbf{Y})$ is calculated from the final outputs \mathbf{y}^N and the label \mathbf{Y} .

2. **Backward pass:** For a given input \mathbf{x} we now consider the partial derivatives of the loss with respect to the weights w_{ij}^l . Hence, we have to apply the chain rule of differentiation.

¹Our algorithm can also get stuck in a local minimum. We will come back to that problem in the context of the parameter update in Sec. 5.3.2

5. Deep Learning

We calculate

$$\begin{aligned}
 \frac{\partial \mathcal{L}(\mathbf{y}^N(x, \mathbf{w}), \mathbf{Y})}{\partial w_{ij}^l} &= \frac{\partial \mathcal{L}}{\partial \mathbf{y}^N} \frac{\partial \mathbf{y}^N}{\partial w_{ij}^l} \\
 &= \frac{\partial \mathcal{L}}{\partial \mathbf{y}^N} \frac{\partial \mathbf{y}^N}{\partial \mathbf{y}^{N-1}} \frac{\partial \mathbf{y}^{N-1}}{\partial w_{ij}^l} \\
 &= \dots \\
 &= \frac{\partial \mathcal{L}}{\partial \mathbf{y}^N} \frac{\partial \mathbf{y}^N}{\partial \mathbf{y}^{N-1}} \dots \frac{\partial \mathbf{y}^{l+1}}{\partial \mathbf{y}^l} \frac{\partial \mathbf{y}^l}{\partial w_{ij}^l}.
 \end{aligned} \tag{5.13}$$

Notice that each time we go to an earlier layer $l - 1$, we get an additional term in this expression. If we calculate the partial derivatives for the weights in the last layer N , we have to determine [152]

$$\frac{\partial \mathcal{L}(\mathbf{y}^N)}{\partial w_{ij}^N} = \boldsymbol{\delta}^N \frac{\partial \mathbf{y}^N}{\partial w_{ij}^N} = \delta_i^N a'^N(z_i^N) y_j^{N-1}, \quad \delta_i^N := \frac{\partial \mathcal{L}}{\partial y_i^N}(\mathbf{y}^N) \tag{5.14}$$

where $a' = \frac{da(z)}{dz}$ is the derivative of the activation function. This derivative is analytically known and \mathbf{z}^N and \mathbf{y}^{N-1} are known from the forward pass. $\boldsymbol{\delta}^N$, sometimes called the error term of layer N [152], is just the analytically known derivative of the loss function \mathcal{L} with respect to the components of \mathbf{y}^N evaluated at the known \mathbf{y}^N values from the forward pass. Hence, calculating this equation requires no additional pass through the network.

If we define $\boldsymbol{\delta}^l$ in the same way for layer l , we get a recursive relationship [152]:

$$\begin{aligned}
 \delta_i^l &:= \frac{\partial \mathcal{L}}{\partial \mathbf{y}^N} \frac{\partial \mathbf{y}^N}{\partial \mathbf{y}^{N-1}} \dots \frac{\partial \mathbf{y}^{l+1}}{\partial y_i^l} \\
 &= \sum_k \delta_k^{l+1} \frac{\partial y_k^{l+1}}{\partial y_i^l} \\
 &= \sum_k \delta_k^{l+1} a'^{l+1}(z_k^{l+1}) w_{ki}^{l+1}.
 \end{aligned} \tag{5.15}$$

In this equation the derivative of the activation function $a'^{l+1}(z)$ is again analytically known and has to be evaluated with the values we get from a forward pass \mathbf{z}^{l+1} . The current weights \mathbf{w}^{l+1} are of course also known. So the only remaining term to calculate is $\boldsymbol{\delta}^{l+1}$. However, this term can be substituted recursively until we get back to eq. (5.14). The gradient (5.13) can therefore be calculated as

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial w_{ij}^l} &= \boldsymbol{\delta}^l \frac{\partial \mathbf{y}^l}{\partial w_{ij}^l} \\
 &= \sum_k (\boldsymbol{\delta}^{l+1} a'^{l+1} \mathbf{w}^{l+1})_k \frac{\partial y_k^l}{\partial w_{ij}^l} \\
 &= (\boldsymbol{\delta}^N a'^N \mathbf{w}^N \dots a'^{l+1} \mathbf{w}^{l+1} a'^l)_i y_j^l,
 \end{aligned} \tag{5.16}$$

where we traced the $\boldsymbol{\delta}^l$'s backwards according to (5.15) giving us a long sequence of matrix products from the end of the network to layer l . For that reason this process is known as *backpropagation*.

The backpropagation algorithm reduces the total number of passes through the network required to calculate the gradient to two. The number of parameters M only influences the speed of a single pass but not the number of passes. This huge saving in computing cost is why backpropagation enabled the study of much bigger neural networks.

5.3.2. Parameter Update

After we determined the gradient, we can update our weights with an *optimizer*. The simplest approach is to repeat the backpropagation algorithm with one input sample after the other, giving:

$$w^{(\text{next step})} = w - \eta \frac{\partial \mathcal{L}}{\partial w}. \quad (5.17)$$

In the deep-learning context this is typically called *stochastic gradient descent (SGD)* with a *learning rate* η . While SGD is very fast in calculating any update and is low in memory consumption, calculating the gradient from single samples introduces a lot of noise as the single sample cannot represent the underlying distribution. For example, the gradient calculated from a single cat image will likely not lead to the best classifier of pictures of many different animals. The gradient updates then point in different direction and partially reverse each other, leading to an inefficient path towards the minimum. Another problem with the randomness of SGD is that it can never settle at the minimum and will fluctuate around it. A solution to that problem is gradually reducing the learning rate, similar to the numeric optimization method *simulated annealing*. On the other hand, this randomness allows the network to leave local minima again, making it more likely to converge to the global minimum [150].

Batch gradient descent (BGD) also follows eq. (5.17) but it uses the average gradient over many input samples. This increases the cost per update and memory usage. However, the larger the batch size, the smaller the random fluctuation in the gradient descent because the batch of data represents the underlying distribution of the training samples better than a single sample. This leads to a shorter path towards the minimum but also increases the risk to get stuck in a suboptimal local minimum. One big advantage of BGD compared to SGD is that BGD can make use of matrix operations which are highly parallelizable and therefore can profit very much from running on GPUs while SGD has to calculate every small descent step sequentially [150]. The balancing of the batch size is an important part of hyperparameter optimization of a network (cf. 5.4.3).

However, regular gradient descent presented here is still quite slow. A faster optimizer is for example momentum optimization. The introduction of a momentum term m increases the chances of avoiding local minima and accelerates convergence to a minimum by building up momentum (see e.g. [150]).

$$w^{(\text{next step})} = w + m \quad (5.18)$$

$$m^{(\text{next step})} = \beta m - \eta \frac{\partial \mathcal{L}}{\partial w}. \quad (5.19)$$

The momentum is usually initialized as $m = 0$. The parameter $\beta \in [0, 1]$ is usually chosen as $\beta \sim 0.9$.

The speed of optimizers however often also depends on careful tuning of the learning rate η . Some popular optimizers, including ADAM [162] and NADAM [163], avoid this by using adaptive learning rate algorithms. In these optimizers the learning rate η is typically reduced over time in order to allow the algorithm to settle down to a loss minimum, often leading to faster and better convergence [150]. The choice of an optimizer is also part of the hyperparameter optimization of a network (cf. 5.4.3).

For the search for continuous gravitational waves with a DNN, we decided to use the adaptive learning rate algorithm ADADELTA [164]. It is very fast and robust and does not require much hyperparameter tuning. ADADELTA was derived from ADAGRAD [165] and works by locally accumulating past gradients as an exponentially decaying average of the squared gradients

$$g^2 = \left(\frac{\partial \mathcal{L}}{\partial w} \right)^2. \quad (5.20)$$

5. Deep Learning

Assuming gradient updates at timesteps $t, t - 1, \dots$ we can write the running average squared gradient $E[g^2]$ as

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho)g_t^2, \quad (5.21)$$

where $E[g^2]_0 = 0$ is the initial value and ρ is a decay constant similar to the β for the momentum method. The weight update then works as

$$w_t = w_{t-1} + \Delta w_t \quad (5.22)$$

with

$$\Delta w_t := -\frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}}g_t, \quad (5.23)$$

where a constant $\varepsilon > 0$ is added to make the denominator more stable. However, the learning rate η in ADADELTA is not constant but also determined as an exponentially decaying running average of the past parameter updates $\Delta w_{t-1}, \dots, \Delta w_0$:

$$E[\Delta w^2]_t = \rho E[\Delta w^2]_{t-1} + (1 - \rho)\Delta w_t^2. \quad (5.24)$$

This running average is used for an approximation of a parameter update with a diagonal Hessian (Newton's method) for locally smooth curvature. The second order parameter update is given by

$$\Delta w \sim \frac{\frac{\partial \mathcal{L}}{\partial w}}{\frac{\partial^2 \mathcal{L}}{\partial w^2}}. \quad (5.25)$$

In the approximation the inverse of the second derivative is calculated as

$$\frac{1}{\frac{\partial^2 \mathcal{L}}{\partial w^2}} \sim \frac{\Delta w}{\frac{\partial \mathcal{L}}{\partial w}}, \quad (5.26)$$

where the Δw in the numerator is calculated with the running median in eq. (5.24) and the gradient $\frac{\partial \mathcal{L}}{\partial w}$ in the denominator from eq. (5.21). Therefore, the total parameter update is performed by

$$\Delta w_t = -\frac{\sqrt{E[\Delta w^2]_{t-1} + \varepsilon}}{\sqrt{E[g^2]_t + \varepsilon}}g_t. \quad (5.27)$$

The ε helps with the initialization of the numerator for $\Delta w_0 = 0$ and ensures progress, even if previous updates did get small. The approximation is always positive ensuring that the update follows the negative gradient at each step. Due to the recursive relationship in eq. (5.27) for Δx_t , the denominator lags behind by one time step. This makes the optimizer more robust to large sudden gradients, as the effective learning rate is decreased by the denominator before the numerator can react [164].

5.4. Main Challenges in Deep-Learning

5.4.1. Preparing Data

Constructing a proper *training set* is very important to get reasonable results with a machine-learning algorithm. At the current state machine-learning algorithms need large amounts of data to learn: thousands of input samples for simple problems and millions of samples for image or speech recognition [150]. In general, it is beneficial to have the training set as big as possible (cf. Sec. 5.4.2). Unfortunately, it is often limited in size by the ability to label data efficiently. In the context of gravitational-wave detection this is usually not an issue as high quality modeled signals, which originally were developed for matched-filtering searches (cf. Sec. 4.3), can be used to generate training samples. In this case the training set size is mainly limited by memory and computing cost considerations.

The training set also works similar to a Bayesian prior distribution (cf. Sec. 4.4.1). The ratio of samples of the different classes sets a weight on their contribution to the loss function, making it easier to classify an overrepresented class but less likely to identify an underrepresented class. For example, if an image classifier is trained on a training set containing only one image of a cow and otherwise cats and dogs, the resulting network will most likely not be able to learn to recognize cows in new images. Therefore, the training data should follow the same distribution as the underlying problem or at least be a subdistribution of it.

Another important property of the training set is the size of the input samples themselves. If they contain many unimportant pieces of information the network will have difficulties extracting the relevant parts. Therefore, it is crucial to condense the relevant information of training set samples to the smallest size possible [150]. In the deep-learning approach to continuous gravitational wave searches (cf. Chs. 7 and 8) this is solved by reducing the input size to the maximal size a signal can have in the frequency domain.

A very important step with regard to the handling of data is to have at least one separate *test set*, which is not used for training or network architecture optimization. A test set can either have the same distribution of samples as the training set, i.e. the same as the problem we are trying to solve, or it can test how the network performs for samples outside the training set distribution. This is called testing the *generalization* of the network. For example, a test could check how a cat-dog classifier would classify images of wolves or in the context of gravitational waves how a signal-noise classifier deals with different signal types or different noise disturbances (cf. Secs. 7.4.1 and 7.4.1).

Carefully constructing the test set is arguably even more important than the construction of the training set. A bad training set gives bad network performance, but a bad test set gives unreliable results. If only the training set itself is used for testing, the network will start to recognize classes from certain exceptional features of the training samples that might be irrelevant to the actual task. For example, recognizing polar bears just from the snowy background. The network will then show very good performance on the training set but much worse performance on the test set. This is called *overfitting* and is discussed in more detail in Sec. 5.4.2.

If we do not want to use a single neural network but find the optimal network for our task, we have to use a third set of data samples: the *validation set*. The validation set should consist of samples from the same distribution as the training set. If the available data is limited this is usually achieved by a so-called validation split: A large set of samples is split in e.g. 80% training samples and 20% validation samples. Extensively optimizing the network architecture according to the performance on the validation set, however, can lead to a bias towards specific features in the validation set. Then the performance on the validation set is very good but the network fails on the test set. This again is a form of overfitting.

Training and validation set results, e.g. loss over time, should only be used to monitor the training and architecture optimization progress, respectively. Only a properly constructed, independent test set can give an unbiased final performance estimate.

5.4.2. Overfitting and Underfitting

During or after training a neural network, you can usually identify bad performance as one of two problems: *overfitting* or *underfitting*. In order to recognize over- and underfitting we need our training set, validation set and test set to all contain different samples from the same distribution.

If a neural network shows significantly better performance on the training set than on the validation set (when optimizing the network architecture: better performance on the validation set than on the test set), this is called *overfitting* the training set (validation set). An example of overfitting is shown for the detection probability of a CW neural network in Fig. 5.6a.

Overfitting typically occurs if we have too many parameters in the network relative to the

5. Deep Learning

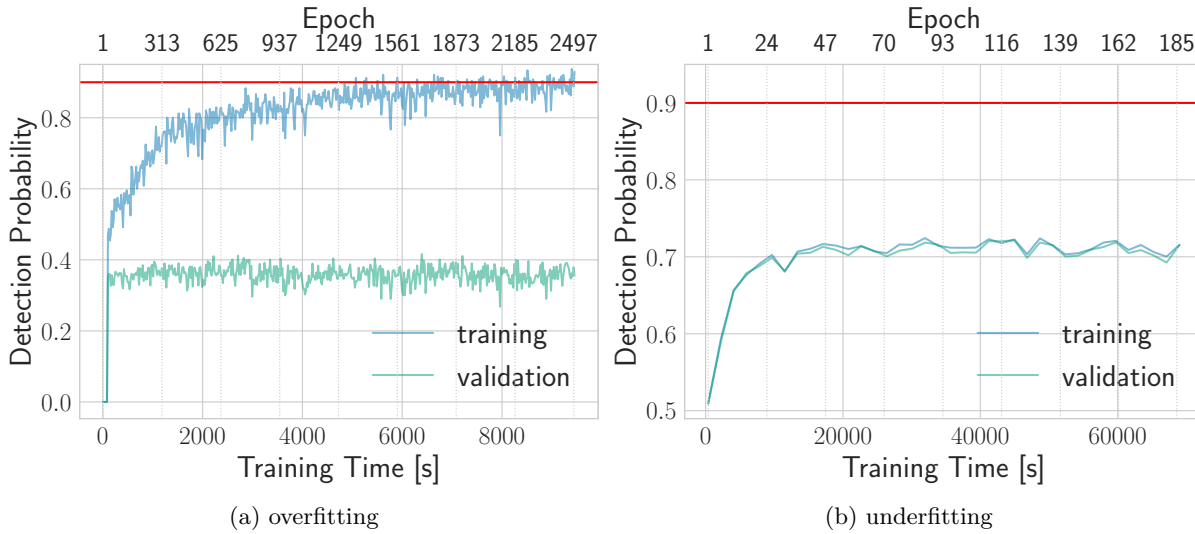


Figure 5.6.: Detection probability vs. training time of a noise-signal classifier for CWs: In (a) the network clearly overfits the data. The training performance reaches the desired performance marked in red and the validation set performance remains low although both sets contain samples of the same distribution. In (b) the network underfits the data. Training and validation performance are very similar but far from the desired performance marked by the red line.

amount of different data samples. The network will start “memorizing” exactly the training samples and loses the ability to generalize. This corresponds to fitting a polynomial of a high order to a low number of data points, see the orange 20th degree polynomial fit in Fig. 5.7. While the fit does give the correct value at every data point, it does not resemble the original underlying function and would give bad predictions if we added additional points.

Overfitting can be avoided by increasing the training set size or by reducing the number of parameters in the network, as well as some other techniques. In the situation where the training set can easily be increased in size, like a CW search, increasing the training set is typically the right way to go. Alternatively, if the network gets so large that it gets difficult and time-consuming to train, shrinking the network is also an option. In the case of limited training data it is common to use so-called *regularization layers* to artificially add additional information to prevent overfitting. For example, Dropout layers [166, 167] add small amounts of damping noise to the network training by randomly setting some weights to zero during the training step. Another layer that has a regularization effect is the Batch Normalization layer that we will discuss in Sec. 5.5.3 [150].

A network can also underfit the training data. In this case the performance of the network on the training set, validation set and test set is similar but suboptimal (see Fig. 5.6b). Underfitting occurs if the network does not have enough parameters for the complexity of the problem or if suboptimal training prevents the network from making use of enough parameters². An example of suboptimal training would be that the gradient descent algorithm settles on a local minimum instead of the global minimum. Underfitting corresponds to fitting a low order polynomial to data points from a higher order polynomial shown by the red cubic fit in Fig. 5.7. Such a fit has large errors for almost all data points.

Underfitting can be fought by increasing the network’s size, e.g. by adding parameters. Additionally, it can be tackled by removing unnecessary complexity from the task to solve, e.g. finding a representation of CW signals such that the input size can be smaller without loos-

²This can also be interpreted as the problem being too hard for the current network to solve.

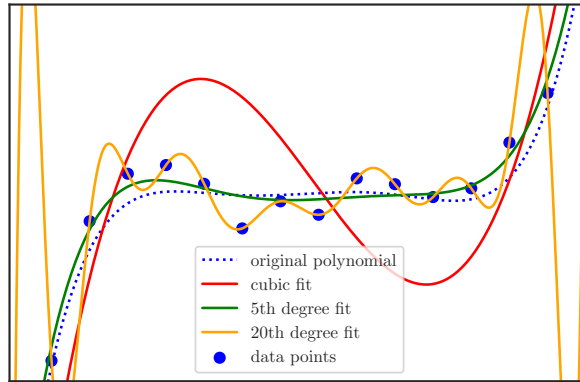


Figure 5.7.: Fitting a fifth degree polynomial (blue, dotted) from some noisy data points (blue) with a cubic polynomial (red), with a 20th degree polynomial (orange) and with a fifth degree polynomial (green). Only the fifth degree polynomial fit approximates the original function with a low error and would give good predictions for newly added data points. The 20th degree polynomial overfits and the third degree polynomial underfits the data.

ing information. If regularization layers are used, removing them can also help improve the performance.

5.4.3. Network Architecture and Hyperparameters

As mentioned in the last section, overfitting and underfitting can be caused by a non-optimal algorithm, i.e. non-optimal network architecture or training. Especially if the available amount of data is limited, improving the algorithm is the only way of avoiding over- and underfitting and improving the network's performance. For that purpose, the algorithm usually contains many *hyperparameters* which can be adjusted to improve performance.

Hyperparameters are parameters that are not determined by the training loop of the neural network, but instead parameters, which determine how well a network trains and performs after training with a certain data set. It includes the number and type of the layers in the network, the types of activation functions used, the weight initialization at the start of the training, the choice of optimizer, the Batch size and the learning rate. The number of possible hyperparameters is very large and it will only be possible to tweak a limited number of parameters. In this work we therefore only briefly discuss a few approaches to optimizing some of the hyperparameters. A more detailed discussion can be found in [150].

The most common approach is manual hyperparameter optimization, where the developer improves the performance by trial and error. Often the success depends on the intuition of the machine-learning developer as there are no fixed rules. Algorithm-based approaches on the other hand require significantly more effort to set up but usually deliver better results [147].

For a long time, the *learning rate* η was arguably considered as the most important hyperparameter [150]. However, as presented in Sec. 5.3.2, some modern *optimizers* like Adadelta [164] are adaptive learning rate algorithms and can achieve similar results as algorithms with hand-picked learning rates. Testing this type of optimizer therefore might bring more performance improvement than tweaking the learning rate.

The *batch size* is often picked as the largest possible size which fits into the memory of the GPU used for training. A larger batch size reduces the noise in the gradient calculation and therefore usually reaches the minimum faster. The choice of batch size is often connected to the learning rate as using small learning rates with large batch sizes can avoid some of the problems of large batch sizes discussed in Sec. 5.3.2 [150].

5. Deep Learning

Most hyperparameters are defined by the *network architecture*. Optimizing all of them can be quite difficult and costly. Therefore, many researchers do not start from scratch, but with an architecture that has shown success in other related fields of research³. These networks can then be tuned and tested on a validation data set for a different problem. It is often useful to start with a network with more parameters than actually needed and prevent overfitting with either regularization or by improving the training set. The reverse, starting with a small network and fighting underfitting, is usually more difficult [150].

As discussed in Sec. 5.2, it can be shown that a neural network can approximate every function if the DNN has enough neurons. However, in many cases fewer neurons are required if they are distributed over more layers. Hence, having a deeper network is often advantageous [155]. On the other hand, deeper networks often have the problem that gradients in early layers tend towards zero. This is called the *vanishing gradient problem* and will be discussed in Sec. 5.4.4.

5.4.4. Vanishing Gradient Problem

An often occurring problem especially in deep neural networks is the *vanishing gradient* problem. As we discussed in section 5.3.1 the network calculates the gradient by recursively calculating error terms δ_l , effectively propagating the error backwards through the network. Due to the many products of eq. (5.16) the gradients for the early layers can get very small if the weights in later layers are smaller than one. Ultimately, the gradient corrections sometimes are so small that no meaningful change in the weights of the earlier layers happens during the gradient descent update. This problem gets worse when a network becomes deeper and deeper, as more and more layers are added [150]. A more detailed study of the reasons for this problem can be found in [158].

This problem was known for a long time and contributed to the partial abandonment of neural networks in the early 2000s. Since then, however, the problem has been partially alleviated when a connection to using saturating activation functions was discovered [158]. By changing from the nature-inspired sigmoid activation function to non-saturating activation functions such as the Rectified Linear Unit function (ReLU) (cf. 5.2) the gradient is kept large for arbitrarily large input values [150].

However, as mentioned in Sec. 5.4.3, it is often advantageous to use deeper networks. Hence, neural networks became deeper and the problem of vanishing gradients reappeared. For networks only consisting of dense layers a permanent solution has been found: the SELU activation function (see [161] and Fig. 5.4) can have a self-normalizing effect in dense networks. For some non-dense architectures, e.g. networks with skip connections or Inception modules (cf. Sec. 5.5.4) as used for the CW search in Chs. 7 and 8, the vanishing gradient problem can still occur and other activation functions might outperform SELU [150].

Popular methods to fight the vanishing gradients problem include normalizing the inputs, *Batch normalization* layers (cf. sec. 5.5.3) and *residual connections* (cf. sec. 5.5.4).

The opposite problem is less common. If many large terms are multiplied, the gradient corrections for the early layers become too large. This is called the *exploding gradients problem* [150]. Besides normalization and Batch normalization, one common approach to fighting this problem is *gradient clipping*, i.e. clipping a gradient if it exceeds a fixed maximum value (see [168]).

³From personal communication of the author at the Deep Learning Bootcamp 2017, Dresden <https://indico.mpi-cbg.de/event/42/overview>

5.5. Advanced Concepts

5.5.1. Convolutional Layer

We often want to detect local features independently of their position in the input data. This translational invariance can be achieved by using *convolutional layers*.

In a convolutional layer the neurons are not connected to every single neuron of the previous layer but are connected only to a small area called the *receptive field*. This receptive field is then slid over the input giving one output neuron per application (see Fig. 5.8a). In a convolutional layer the weights are shared between neurons, i.e. we construct a *filter* consisting of one weight for every neuron in the receptive field. If we consider multiple convolutional layers the receptive field of the last layer in the first input grows with every layer (see Fig. 5.8d). Sometimes the receptive field of a single layer is shifted by more than one neuron from one output neuron to the next to spread out the receptive field even more. The step size is usually called *stride*.

Due to the limited receptive field, single convolutional layers can only detect small low-level features, e.g. edges or corners, in the receptive field. However, sliding the filter makes sure that the network learns to detect a certain feature independently of its position in the input. Stacking these layers leads to a hierarchical combination of smaller features to large scale features, e.g. geometrical shapes, faces, cars, animals, etc. [150]. With each additional convolutional layer the receptive field with respect to the input data is increased. Stacking enough convolutional layers therefore allows contributions from the entire input in a single filter (see Fig. 5.8d). Furthermore, because a certain feature is detected independently of its position, a network with enough convolutional layers will often have a translational invariance of features. This is very important in image recognition in order to recognize e.g. a cat at any position in the image. It is also relevant for our use case of CW searches as it ensures that signals will be detected by their shape no matter the exact position in the frequency band (cf. Sec. 7.3).

The sliding of the receptive field also allows fully convolutional networks to not require a fixed input size. However, this ability breaks down as soon as a dense layer is added to the network. Nevertheless, many networks, e.g. AlexNet [169] and ResNet[170], make use of dense layers at the end because they are good at classifying the data dependent on the features detected by convolutional layers.

A single filter would only allow a convolutional layer to detect a single feature. Instead a convolutional layer uses multiple filters producing a respective output for every single one of them. These outputs are called *feature maps* as they encode where in the input the different features can be found. Technically these feature maps are stored in an additional dimension with the size equaling the number of filters. This is the *channels* dimension of the neural network introduced in Sec. 5.2. If the input to a convolutional layer already possesses multiple channels, e.g. a color picture with RGB channels, the convolutional filters always extend over the entire channels dimension (see Fig. 5.8c), thereby relating features from different channels.

The filter size and the number of filters fix the number of weights in a convolutional layer, making it independent of the number of neurons in the previous layer. In a dense network every added neuron in the input adds weights for every neuron in the previous dense layer: the number of weights grows linearly with the number of neurons. The number of weights in a convolutional layer

$$N_{\text{weights}}^{\text{Conv}} = c \cdot f \quad (5.28)$$

is just the product of the number of feature maps or channels c and the size of the filter f . Hence, a convolutional neural network can cover a large input with far less parameters than a dense network, making it computationally cheaper. However, convolutional layers rely on the data to be separable in features. Dense networks on the other hand are input agnostic.

The output of a convolutional layer is typically smaller than the input size, as the receptive field cannot be centered on the edges of the input shape. To keep the output size the same as the input size, the input is usually zero-padded to avoid these size losses at the edge (see Fig. 5.8b).

5. Deep Learning

If a stride > 1 is used, the output will be even further decreased in size. This is usually done intentionally to reduce the computational cost of subsequent layers. However, in most cases this task can be fulfilled better by *pooling layers*.

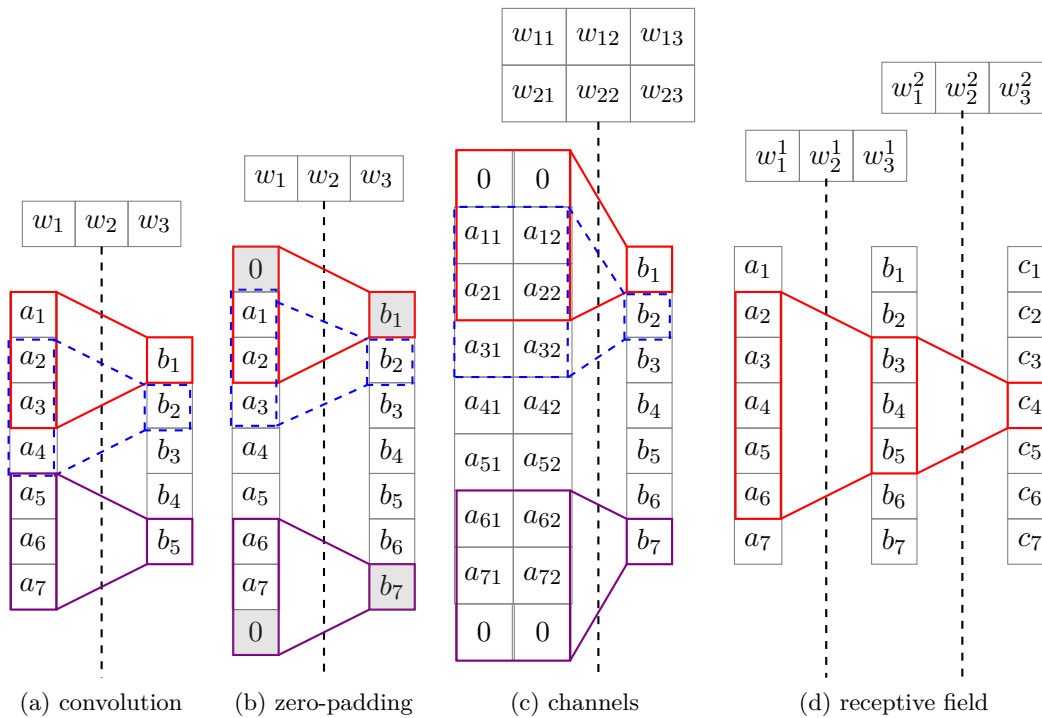


Figure 5.8.: Application of a size 3 filter in a convolutional layer: In a simple convolutional layer (a) the filter is moved over the input a , giving a smaller output b . Often the edges are zero-padded (b) to keep the output the same size as the input. If the input has multiple channels (two channels in example (c)) the filter has three weights per channel and the receptive field extends over all channels. If multiple convolutional layers are stacked (e.g. two filter size 3 layers in (d)), the network's receptive field, measured on the input a , grows.

5.5.2. Pooling Layers

A Pooling layer is similar to a convolutional layer in the sense that it possesses a receptive field which moves over the input data for the different output neurons. However, instead of learning a filter, i.e. a linear transformation, it performs a fixed operation: typically either taking the maximum or the average over the receptive field [147]. Pooling layers usually work with a stride such that the receptive fields of the neurons do not overlap.

A common problem for convolutional layers is the large amount of memory required to store values calculated during the forward pass and needed for the backpropagation algorithm [150]. The convolutional layer is well suited for extracting features from data but it usually only slightly decreases the data in size and sometimes even enlarges the data by adding more channels. Pooling layers are therefore introduced to downsample the input such that following convolutional layers are computationally more efficient [150]. Reducing the size of the network in that way allows for more stacked layers and therefore often improves performance [147].

Furthermore, pooling layers help a convolutional network to be locally translational invariant. A pooling layer combines neighbouring input neurons in a feature map essentially keeping only the information that the feature is in any of the inputs covered by its receptive field [150].

The most common type of pooling layer described here is applied channel by channel only reducing the input dimensions. A different type, so-called depth-wise pooling layers, specifically pool different features together, combining for example rotated versions of the same feature. For a similar purpose, convolutional neural networks sometimes use convolutional *bottleneck* layers: convolutional layers with a filter size of one and a smaller number of channels than the input. This forces the network to combine feature maps with a weighted sum, creating higher level features. We will come back to these in Sec. 5.5.4.

5.5.3. Normalization and Batch Normalization

An important step to successfully train a neural network is input normalization. If the input is not normalized, the different input samples often lead to very differently scaled weight derivatives. This can produce conflicting gradients slowing down the learning process significantly. While input normalization fixes this problem for the input, a similar problem can occur between any two layers of the network.

Batch normalization introduced by Ioffe and Szegedy in 2015 [171] adds a normalization layer just before or after all or some of the activation functions in the network. It allows the network to learn the optimal normalization for the inputs of each layer in the network. Batch Normalization helps with gradient propagation through the network. The vanishing gradient problem of Sec. 5.4.4 can be fought quite successfully with this technique.

A batch normalization layer introduces two trainable parameters β and γ per channel, which are the optimal mean and scale of the output. The layer first zero-centers and normalizes the input x with the estimated input mean μ_B and standard deviation σ_B calculated over all input dimensions and over a batch of input samples.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (5.29)$$

where ϵ is a small constant preventing divisions by zero and the indices i numerate the samples in the batch of data. The normalized input \hat{x} is then rescaled and offset by the learned parameters setting the mean and standard deviation of the output to β and γ , respectively:

$$y_i = \gamma \hat{x}_i + \beta. \quad (5.30)$$

During training Batch normalization calculates the mean μ_B and standard deviation σ_B over a batch of data. However, when we apply a network, we can in general not expect to have a batch of data. We might only have a single data sample and hence cannot calculate the mean and standard deviation in the presented way. Thus, the Batch Normalization layer possesses additional learned parameters determined during training but separately from the usual back-propagation algorithm: a moving mean μ and standard-deviation σ of the input replacing the batch mean and standard deviation. It is updated after every batch as

$$\mu' = m\mu + (1 - m)\mu_B \quad (5.31)$$

$$\sigma' = \sigma^2 + (1 - m)\sigma_B^2 \quad (5.32)$$

with momentum term $m \approx 0.99$ [150].

5.5.4. Residual Connections and Inception Modules

Two additional concepts became popular after causing significant improvements in the image classification on the ImageNet data set [172]: *Inception* [173] and *residual* [170] networks. They won the ILSVRC contest⁴ in 2014 and 2015 respectively.

⁴The Imagenet large scale visual recognition challenge [174] was an annually conducted contest from 2010 to 2017 which looked for the best performing algorithm in various visual recognition tasks, especially on the ImageNet data set.

5. Deep Learning

Residual networks introduce residual connections, also called skip or shortcut connections, in the network. These are connections of non-neighbouring layers. For example, they may connect a layer l directly to the layer $l + 2$ (see Fig. 5.9a). This residual connection adds the output of the previous layer to the output of the circumvented part of the network and applies the activation function after the sum. Let us assume without the skip connection layer $l + 1$ in Fig. 5.9a approximates a function $T(y^l)$. If we now add the skip connection passing layer $l + 2$ the approximated function changes to [150]

$$\bar{T}(y^l) = T(y^l) + y^l. \quad (5.33)$$

Especially after initialization layers often have very small weights and an output close to zero. A layer with a residual connection reproduces the input instead. If we look at the gradient calculation for the weights of layer l , the derivative

$$\frac{\partial \bar{T}}{\partial w^l} = \frac{\partial T}{\partial w^l} + \frac{\partial y^l}{\partial w^l} = \frac{\partial T}{\partial w^l} + a'(z^l)y^{l-1} \quad (5.34)$$

does not vanish even if $\frac{\partial T}{\partial w^l}$ does. Therefore, if in a non-residual network the gradient for w^l vanishes when we propagate it through layer $l + 1$ (cf. Secs. 5.4.4 and 5.3.1), this cannot happen with the skip connection. It allows the gradient to “flow around” those parts of the network that diminish the gradient, making sure that the layers earlier in the network can still be trained. Therefore, residual connections allow to train deeper networks and usually speed up the training as well [170].

Inception modules or blocks were introduced in [173]. Inception modules have layers working in parallel instead of sequential and usually replace simple convolutional layers. Instead of a single convolutional layer with a fixed filter size, the Inception module trains and calculates multiple layers from a shared input. Each layer calculated from the initial input starts its own sub-sequence of layers, which we will call *columns* (see Fig. 5.9b). Typically in the columns the Inception module contains convolutional layers with different filter sizes and pooling layers (see Fig.5.9b). After a couple of layers in each column the column outputs are recombined by concatenating the resulting feature maps in the channels dimension.

Hereby, zero-padding ensures (see Fig. 5.8b) that the feature maps of the different convolutional layers have the same input-dimension sizes and can be concatenated. The different filter sizes in the columns allow the network to learn differently-scaled features. Furthermore, the different paths in the Inception block often contain additional convolutional layers with size one. These filters cannot identify features along the input dimensions but only along the channel dimension. They are usually used to combine different features and reduce the size of the data. Therefore, each of the differently sized convolutional layers in the Inception module can have many feature maps but the combined output of the Inception module is kept relatively small. The construction of the Inception module makes training a network with the same number of parameters as a sequential convolutional network much more efficient and effective. Hence, it is also possible to make deeper networks with Inception module, improving performance further [150].

The combination of residual connections and Inception modules brought further improvements in the field of image recognition in the form of the InceptionResnet architecture [175]. The networks developed in the context of this work to search for continuous gravitational waves make use of both: residual connections and Inception modules. They are 1D adaptations of the 2D networks presented in [170] and [175]. One InceptionResnet block used in Ch. 8 is illustrated in Fig. 5.10.

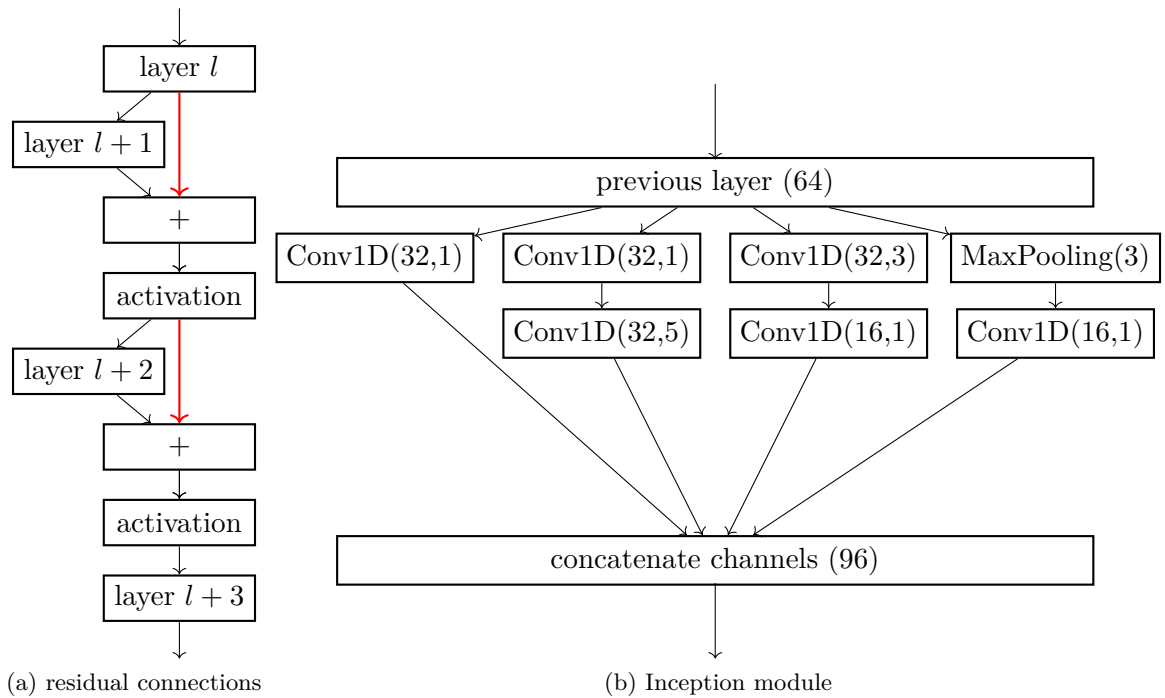


Figure 5.9.: (a) Residual connections (red) allow the network to skip for example the layers $l+1$ or $l+2$. The gradient calculation is therefore less affected by the weights of the circumvented layers improving the gradient calculation for earlier layers. Inception modules (b) use parallel layers with different filter sizes to allow more diverse features to be detected. In this figure the notation Conv1D(c, f) denotes a 1D convolutional layer with a filter size of f , c feature maps and a stride of 1. MaxPooling(3) is a max pooling layer with a receptive field of 3 and a stride of 3. In the end the columns in the Inception module are combined by concatenation resulting in 96 channels. The previous layer had 64 channels.

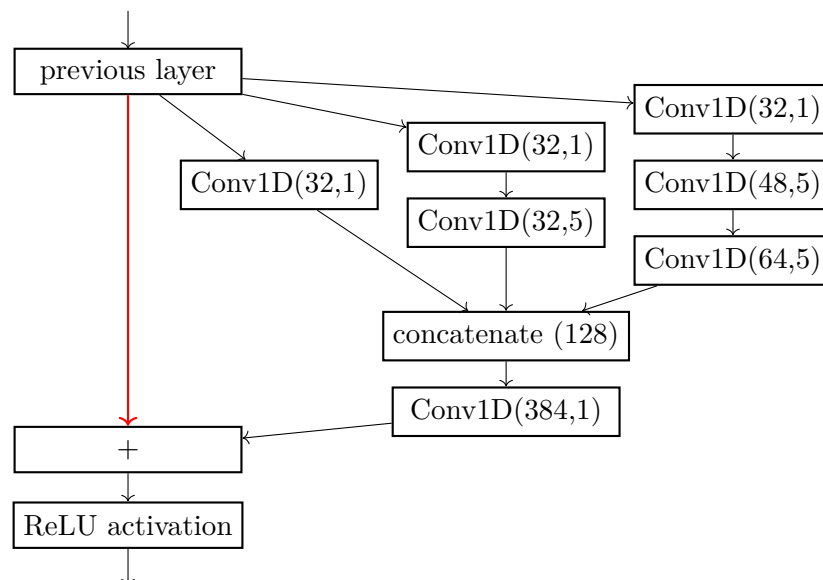


Figure 5.10.: Inception-ResNet-A block as used in Ch. 8 with the same notation as in Fig. 5.9. The residual connection (red) skips the entire Inception module. Compared to the original block (see Fig. 16 in [175]) the 3×3 filters were replaced by one dimensional filters of size 5. The other InceptionResNet-v2 blocks were adapted similarly.

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

This chapter is essentially a reprint of the paper published as Dreissigacker, Prix, and Wette [28], with minor changes and reformatting as a chapter of this thesis.

6.0. Abstract

This chapter presents an efficient numerical sensitivity-estimation method and implementation for continuous-gravitational-wave searches, extending and generalizing an earlier analytic approach by Wette [29]. This estimation framework applies to a broad class of \mathcal{F} -statistic-based search methods, namely (i) semi-coherent StackSlide \mathcal{F} -statistic (single-stage and hierarchical multi-stage), (ii) Hough number count on \mathcal{F} -statistics, as well as (iii) Bayesian upper limits on \mathcal{F} -statistic search results (coherent or semi-coherent). We test this estimate against results from Monte-Carlo simulations assuming Gaussian noise. We find the agreement to be within a few % at high detection (i.e. low false-alarm) thresholds, with increasing deviations at decreasing detection (i.e. higher false-alarm) thresholds, which can be understood in terms of the approximations used in the estimate. We also provide an extensive summary of sensitivity depths achieved in past continuous-gravitational-wave searches (derived from the published upper limits). For the \mathcal{F} -statistic-based searches where our sensitivity estimate is applicable, we find an average relative deviation to the published upper limits of less than 10%, which in most cases includes systematic uncertainty about the noise-floor estimate used in the published upper limits.

6.1. Introduction

The recent detections of gravitational waves from merging binary-black-hole and double neutron-star systems [1, 2, 176] have opened a whole new observational window for astronomy, allowing for new tests of general relativity [64], new constraints on neutron star physics [82] and new measurements of the Hubble constant [4], to mention just a few highlights.

Continuous gravitational waves (CWs) from spinning non-axisymmetric neutron stars represent a different class of potentially-observable signals [85, 177], which have yet to be detected [91]. These signals are expected to be long-lasting (at least several days to years) and quasi monochromatic, with slowly changing (intrinsic) frequency. The signal amplitude depends on the rich (and largely not yet well-understood) internal physics of neutron stars [178], as well as their population characteristics [78, 179]. A detection (and even non-detection) of CWs could therefore help us better understand these fascinating astrophysical objects, and may allow for new tests of general relativity [67, 68].

Overview of Search Categories

We can categorize CW searches in two different ways: either based on the *search method*, or on the type of *explored parameter space*.

The *search methods* fall into two broad categories: *coherent* and *semi-coherent* (sometimes also referred to as *incoherent*). Roughly speaking, a coherent search is based on signal templates with coherent phase evolution over the whole observation time, while semi-coherent searches typically break the data into shorter coherent segments and combine the resulting statistics

from these segments incoherently (i.e. without requiring a consistent phase evolution across segments). However, there are many different approaches and variations, which are beyond the scope of this work, see e.g. [91] for a more detailed overview. Here we will exclusively focus on coherent and semi-coherent methods based on the \mathcal{F} -statistic, which will be introduced in Sec. 6.2.

Coherent search methods are the more sensitive in principle, but in practice they suffer from severe computing-cost limitations: For finite search parameter spaces the required number of signal templates grows as a steep power-law of the observation time, making such searches infeasible except when the search region is sufficiently small. For larger signal parameter spaces the observation time needs to be kept short enough for the search to be computationally feasible, which limits the attainable coherent sensitivity. This is where semi-coherent searches yield substantially better sensitivity at fixed computing cost (e.g. see [126, 180]).

Based on the *explored parameter space*, we distinguish the following search categories (referencing a recent example for each case):

- (i) *Targeted searches* are looking for CW emission from known pulsars [181]. The pulsar spin evolution, the sky position and possibly the binary orbital parameters of these systems are known very accurately. Assuming a fixed relationship between pulsar spin and CW frequency, these searches therefore only need to target a single point in parameter space for each pulsar. Hence these searches are done with an optimal fully-coherent search [182].
- (ii) *Narrow-band searches* for known pulsars assume a small uncertainty in the relationship between CW frequency and the measured pulsar spin rates. This finite search parameter space requires a template bank with (typically) many millions of templates, still allowing for optimal fully-coherent search methods to be used [183].
- (iii) *Directed (isolated) searches* aim at isolated neutron stars with known sky-position and unknown spin frequency. The search parameter space covers the unknown frequency and spindowns of the neutron star signal within an astrophysically-motivated range [184, 185].
- (iv) *(Directed) binary searches* aim at binary systems with known sky-position and parameter-space uncertainties in the frequency and binary-orbital parameters. Typically these sources would be in low-mass X-ray binaries, with the most prominent example being Scorpius X-1 (Sco X-1) [135, 186].
- (v) *All-sky (isolated) searches* search the whole sky over a large frequency (and spindown) band for unknown isolated neutron stars [187, 188].
- (vi) *All-sky binary searches* are the most extreme case, covering the whole sky for unknown neutron stars in binary systems [100, 189].

Sensitivity Estimation

In this work we use the term *sensitivity* to refer to the upper limit on *signal amplitude* h_0 (or equivalently *sensitivity depth* $\mathcal{D} \equiv \sqrt{S}/h_0$, see Sec. 6.2.5). This can be either the frequentist upper limit for a given detection probability at a fixed false-alarm level (p-value), or the Bayesian upper limit at a given credible level for the given data.

Sensitivity therefore only captures one aspect of a search, namely how “deep” into the noise-floor it can detect signals, without accounting for how “wide” a region in parameter space is covered, how much prior weight this region contains, or how robust the search is to deviations from the signal model. Comparing sensitivity depth therefore only makes sense for searches over very similar parameter spaces. A more complete measure characterizing searches would be their respective detection probability, which folds in sensitivity depth, breadth in parameter space, as well as the prior weight contained in that space [190, 191].

However, it is often useful to be able to reliably and cheaply estimate the sensitivity of a search setup without needing expensive Monte-Carlo simulations:

- In order to determine optimal search parameters for a semi-coherent search (i.e. the number and semi-coherent segments and template-bank mismatch parameters), it is important to be able to quickly assess the projected sensitivity for any given search-parameter combination (e.g. see [180, 190–192]).
- For setting upper limits for a given search, one typically has to repeatedly add software-generated CW signals to the data and perform a search, in order to measure how often these signals are recovered above a given threshold. By iterating this procedure one tries to find the weakest signal amplitude that can be recovered at the desired detection probability (or “confidence”). This can be very computationally expensive, and a quick and reasonably-reliable estimate for the expected upper-limit amplitude can therefore substantially cut down on the cost of this iterative process, which can also improve the accuracy of the upper limit.
- The estimate can also serve as a sanity check for determining upper limits¹.

A number of theoretical sensitivity estimates have been developed over the past decades. One of the first estimates was obtained for a coherent \mathcal{F} -statistic search [193], yielding

$$h_0 = 11.4 \sqrt{\frac{\mathcal{S}}{T_{\text{data}}}}, \quad (6.1)$$

for a 90% confidence upper limit at 1% false-alarm (per template). \mathcal{S} denotes the (single-sided) noise power spectral density, and T_{data} is the total duration of data from all detectors, e.g. for a search spanning one day (i.e. $T_{\text{span}} = 24$ h), one detector could have yielded 18 h of (possibly non-contiguous) usable data and another detector 16 h, giving a total of $T_{\text{data}} = 34$ h.

This first estimate was later generalized to the semi-coherent Hough [130] and StackSlide method [194, 195], yielding an expression of the form

$$h_0 = \kappa N_{\text{seg}}^{1/4} \sqrt{\frac{\mathcal{S}}{T_{\text{data}}}}, \quad \text{with } \kappa \sim 7 - 9, \quad (6.2)$$

for the same confidence and false-alarm level as Eq. (6.1), and where N_{seg} denotes the number of semi-coherent segments.

These latter results suggested the inaccurate idea that the sensitivity of semi-coherent searches follows an exact $N_{\text{seg}}^{1/4}$ scaling. However, this was later shown [29, 180] to not be generally a good approximation except asymptotically in the limit of a large number of segments ($N_{\text{seg}} \gtrsim 100 - 1000$).

Furthermore, these past sensitivity estimates relied on the assumption of a “constant signal-to-noise ratio (SNR)” population of signals. While this approximation substantially simplifies the problem, it introduces a noticeable bias into the estimate, as discussed in more detail in [29]. For example, the constant-SNR bias combined with the incorrect $N_{\text{seg}}^{1/4}$ scaling in Eq. 6.2 would result in an overestimate by a *factor of two* of the sensitivity of the first Einstein@Home search on LIGO S5 data [196].

These limitations of previous sensitivity estimates were eventually overcome by the analytic sensitivity-estimation method developed by Wette for semi-coherent StackSlide \mathcal{F} -statistic searches. In this work we simplify and extend this framework by employing a simpler direct numerical implementation. This further improves the estimation accuracy by requiring

¹In fact, in the course of this work we have identified a bug in the upper-limit script of a published result, while trying to understand the discrepancy between the estimate and the published value, see Sec. 6.6.3.

fewer approximations. It also allows us to generalize the framework to multi-stage hierarchical StackSlide- \mathcal{F} searches, Hough- \mathcal{F} searches (such as [196]), as well as to Bayesian upper limits based on \mathcal{F} -statistic searches.

Plan of this Chapter

Sec. 6.2 provides a description of the CW signal model and introduces different \mathcal{F} -statistic-based search methods. In Sec. 6.3 we present the sensitivity-estimation framework and its implementation, for both frequentist and Bayesian upper limits. Section 6.4 discusses how (frequentist) upper limits are typically measured using Monte-Carlo injection-recovery simulations. Section 6.5 provides comparisons of our sensitivity estimates to simulated upper limits in Gaussian noise, while in Sec. 6.6 we provide a comprehensive summary of published sensitivities of past CW searches (translated into sensitivity depth), and a comparison to our sensitivity estimates where applicable. We summarize and discuss the results in Sec. 6.7. Further details on the referenced searches and upper limits are given in appendix 6.8. More technical details on the signal model can be found in appendix 6.9. Finally, appendix 6.11 contains a discussion of the distribution of the maximum \mathcal{F} -statistic over correlated templates.

6.2. \mathcal{F} -statistic-based Search Methods

This section provides an overview of the \mathcal{F} -statistic-based search methods for which sensitivity estimates are derived in Sec. 6.3. Further technical details about the signal model and the \mathcal{F} -statistic are given in appendix 6.9. For a broader review of the CW signal model, assumptions and search methods, see for example [85, 91, 177]

6.2.1. Signal Model

For the purpose of sensitivity estimation we assume the data time series $x^X(t)$ from each detector X to be described by Gaussian noise, i.e. $n^X(t) \sim \text{Gauss}(0, \mathcal{S}^X)$ with zero mean and (single-sided) power-spectral density (PSD) \mathcal{S}^X . A gravitational-wave signal creates an additional strain $h^X(t)$ in the detector, resulting in a time series

$$x^X(t) = n^X(t) + h^X(t). \quad (6.3)$$

For continuous gravitational waves the two polarization components can be written as

$$\begin{aligned} h_+(\tau) &= A_+ \cos(\phi(\tau) + \phi_0), \\ h_\times(\tau) &= A_\times \sin(\phi(\tau) + \phi_0), \end{aligned} \quad (6.4)$$

where $\phi(\tau)$ describes the phase evolution of the signal in the source frame. For the quasi-periodic signals expected from rotating neutron stars, this can be expressed as a Taylor series expansion around a chosen reference time (here $\tau_{\text{ref}} = 0$ for simplicity) as

$$\phi(\tau) = 2\pi(f\tau + \frac{1}{2}\dot{f}\tau^2 + \dots), \quad (6.5)$$

in terms of derivatives of the slowly-varying intrinsic CW frequency $f(\tau)$. For a triaxial neutron star spinning about a principal axis, the two polarization amplitudes are given by

$$A_+ = \frac{1}{2}h_0(1 + \cos^2\iota), \quad A_\times = h_0 \cos\iota, \quad (6.6)$$

in terms of the angle ι between the line of sight and the neutron star rotation axis and the overall signal amplitude h_0 . This definition uses the gauge condition of $A_+ \geq |A_\times|$. After translating

the source-frame signal into the detector frame (see appendix 6.9 for details), the strain signal $h^X(t)$ at each detector X can be expressed in the factored form

$$h^X(t; \mathcal{A}, \lambda) = \sum_{\mu=1}^4 \mathcal{A}^\mu h_\mu^X(t; \lambda), \quad (6.7)$$

which was first shown in [102], and where the four amplitudes \mathcal{A}^μ depend on the *amplitude parameters* $\{h_0, \cos \iota, \psi, \phi_0\}$ as given in Eq. 6.60). The four basis functions $h_\mu^X(t; \lambda)$, which are given explicitly in Eq. (6.61), depend on the *phase-evolution parameters* $\lambda = \{\hat{n}, f, \dot{f}, \dots\}$, namely sky position \hat{n} , frequency f and its derivatives $f^{(k)} = d^k f / d\tau^k \big|_{\tau_{\text{ref}}}$, and binary-orbital parameters in the case of a neutron star in a binary.

6.2.2. Coherent \mathcal{F} -statistic

For pure Gaussian-noise time series $\{n^X(t)\}$ in all detectors X , the likelihood can be written as (e.g. see[116–118]):

$$P(x = n | \mathcal{S}) = \kappa e^{-\frac{1}{2}(n, n)}, \quad (6.8)$$

in terms of the multi-detector scalar product

$$(x, y) \equiv 4 \operatorname{Re} \sum_X \int_0^\infty \frac{\tilde{x}^X(f) \tilde{y}^{X*}(f)}{\mathcal{S}^X(f)} df, \quad (6.9)$$

where $\tilde{x}(f)$ denotes the Fourier transform of $x(t)$, and x^* denotes complex conjugation of x . Using the additivity of noise and signals (cf. Eq. (6.3)), we can express the likelihood for data x , assuming Gaussian noise plus a signal $h(\mathcal{A}, \lambda)$ as

$$\begin{aligned} P(x | \mathcal{S}, \mathcal{A}, \lambda) &= P(x - h(\mathcal{A}, \lambda) | \mathcal{S}) \\ &= \kappa e^{-\frac{1}{2}((x-h), (x-h))}. \end{aligned} \quad (6.10)$$

From this we obtain the log-likelihood ratio between the signal and noise hypotheses as

$$\begin{aligned} \ln \Lambda(x; \mathcal{A}, \lambda) &\equiv \ln \frac{P(x | \mathcal{S}, \mathcal{A}, \lambda)}{P(x | \mathcal{S})} \\ &= (x, h) - \frac{1}{2} (h, h). \end{aligned} \quad (6.11)$$

Analytically maximizing the log-likelihood ratio over \mathcal{A} (c.f. appendix 6.9) yields the \mathcal{F} -statistic [102]:

$$\mathcal{F}(x; \lambda) \equiv \max_{\mathcal{A}} \ln \Lambda(x; \mathcal{A}, \lambda) \quad (6.12)$$

The statistic $2\mathcal{F}$ follows a χ^2 -distribution with four degrees of freedom and non-centrality ρ^2 ,

$$P(2\mathcal{F} | \rho^2) = \chi_4^2(2\mathcal{F}; \rho^2), \quad (6.13)$$

with expectation and variance

$$E[2\mathcal{F}] = 4 + \rho^2, \quad \operatorname{var}[2\mathcal{F}] = 8 + 4\rho^2, \quad (6.14)$$

where ρ corresponds to the signal-to-noise ratio (SNR) of coherent matched filtering.

In the *perfect-match* case, where the template phase-evolution parameters λ coincide with the parameters λ_s of a signal in the data x , the SNR can be explicitly expressed as

$$\rho_0^2 \equiv (h, h) = \frac{4}{25} \frac{h_0^2}{\mathcal{S}} T_{\text{data}} R^2(\theta), \quad (6.15)$$

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

where T_{data} is the total duration of data from all detectors², \mathcal{S} denotes the multi-detector noise floor (defined in Eq. (6.19) and (6.20)), and $R(\theta)$ is a geometric factor quantifying the detector response.

The response function $R(\theta)$ (following the definition in [29]) depends on the subset of signal parameters

$$\theta \equiv \{\hat{n}, \cos \iota, \psi\}, \quad (6.16)$$

and is defined with the normalization:

$$\langle R^2 \rangle_{\theta} = 1. \quad (6.17)$$

The explicit expression of R^2 can be found in appendix 6.10. Using this normalization with Eq. (6.15) we can recover the well-known sky- and polarization-averaged squared-SNR expression (e.g. see [102]):

$$\langle \rho_0^2 \rangle_{\theta} = \frac{4}{25} \frac{h_0^2}{\mathcal{S}} T_{\text{data}}. \quad (6.18)$$

The multi-detector noise-floor \mathcal{S} is defined as the *harmonic mean* over the per-detector PSDs \mathcal{S}^X , namely

$$\frac{1}{\mathcal{S}} \equiv \frac{1}{N} \sum_X \frac{1}{\mathcal{S}^X}. \quad (6.19)$$

Note that in practice CW searches do not assume stationary noise over the whole observation time T_{span} , but only over short durations of order ~ 30 mins. This corresponds to the length of the Short Fourier Transforms (SFTs) that are typically used as input data. The present formalism can straightforwardly be extended to allow for this type of non-stationarity [197]. In this case the definition Eq. (6.19) of the multi-detector noise-PSD \mathcal{S} generalizes to the *harmonic mean over all SFTs*,

$$\frac{1}{\mathcal{S}} \equiv \frac{1}{N_{\text{SFT}}} \sum_{\alpha} \frac{1}{\mathcal{S}^{\alpha}}, \quad (6.20)$$

where α is an index enumerating all SFTs (over all detectors), and \mathcal{S}^{α} is the corresponding noise PSD estimated for SFT α .

6.2.3. Semi-coherent \mathcal{F} -statistic Methods

Semi-coherent methods [126] divide the data into N_{seg} segments each spanning a duration $T_{\text{seg}} < T_{\text{span}}$. The segments are analyzed coherently, and the per-segment detection statistics are combined incoherently. Generally this yields lower sensitivity for the same amount of data analyzed than a fully-coherent search. However, the computational cost for a fully-coherent search over the same amount of data is often impossibly large, while the semi-coherent cost can be tuned to be affordable and hence ends up being more sensitive at fixed computing cost [126, 129, 180].

There are a number of different semi-coherent methods currently in use, such as PowerFlux, FrequencyHough, SkyHough, TwoSpect, CrossCorr, Viterbi, Sideband, loosely-coherent statistics and others (e.g. see [91] and references therein). Many of these methods work on short segments, typically of length $T_{\text{seg}} \sim 30$ min, and directly use the power in the frequency bins of Short Fourier Transforms (SFTs) as the coherent base statistic.

In this work we focus exclusively on sensitivity estimation of \mathcal{F} -statistic-based methods, namely StackSlide- \mathcal{F} (e.g. see [180]) and Hough- \mathcal{F} introduced in [130]. Here the length of segments is only constrained by the available computing cost, and segments will typically span many hours to days, which yields better sensitivity, but also requires higher computational cost.

²Not to be confused with the observation time T_{span} , denoting the total time between the first sample of the data and the last.

Therefore, many of the computationally expensive semi-coherent \mathcal{F} -statistic searches are run on the distributed Einstein@Home computing platform [7].

Note that these methods, which use multiple SFTs for every segment, are not to be confused with the (albeit closely related) “classical” StackSlide and Hough methods, which use single SFTs directly as coherent segments, as described for example in [195].

StackSlide- \mathcal{F} : Summing \mathcal{F} -statistics

The StackSlide- \mathcal{F} method uses the sum of the coherent per-segment $\tilde{\mathcal{F}}$ -statistic values in a given parameter-space point λ as the detection statistic, namely

$$2\hat{\mathcal{F}} \equiv \sum_{\ell=1}^{N_{\text{seg}}} 2\tilde{\mathcal{F}}_{\ell}, \quad (6.21)$$

where $\tilde{\mathcal{F}}_{\ell}$ is the coherent \mathcal{F} -statistic of Eq. (6.12) in segment ℓ . This statistic follows a χ^2 -distribution with $4N_{\text{seg}}$ degrees of freedom and non-centrality ρ^2 , i.e.

$$P(2\hat{\mathcal{F}} \mid \rho^2) = \chi_{4N_{\text{seg}}}^2(2\hat{\mathcal{F}}; \rho^2), \quad (6.22)$$

where the non-centrality ρ^2 is identical to the expression for the *coherent squared SNR* of Eq. (6.15), with T_{data} referring to the whole data set used, and \mathcal{S} is the corresponding noise floor. However, the non-centrality in the semi-coherent case cannot be considered a “signal to noise ratio”, due to the larger N_{seg} -dependent degrees of freedom of the χ^2 distribution compared to Eq. (6.13), which increases the false-alarm level at fixed threshold and reduces the “effective” semi-coherent SNR to $\hat{\text{SNR}}^2 = \rho^2 / \sqrt{N_{\text{seg}}}$ (e.g. see Eq.(14) in [198]).

The expectation and variance for $2\hat{\mathcal{F}}$ are

$$E[2\mathcal{F}] = 4N_{\text{seg}} + \rho^2, \quad \text{var}[2\mathcal{F}] = 8N_{\text{seg}} + 4\rho^2. \quad (6.23)$$

We note that StackSlide- \mathcal{F} searches often quote the average $\bar{\mathcal{F}}$ over segments instead of the sum $\hat{\mathcal{F}}$, i.e.

$$\bar{\mathcal{F}} \equiv \frac{1}{N_{\text{seg}}} \hat{\mathcal{F}}. \quad (6.24)$$

Hough- \mathcal{F} : Summing Threshold Crossings

The Hough- \mathcal{F} method [130] sets a threshold $\tilde{\mathcal{F}}_{\text{th}}$ on the per-segment coherent $\tilde{\mathcal{F}}$ -statistics and uses the number of threshold-crossings over segments as the detection statistic, the so-called *Hough number count* n_c , i.e.

$$n_c \equiv \sum_{\ell=1}^{N_{\text{seg}}} \Theta(\tilde{\mathcal{F}}_{\ell} - \tilde{\mathcal{F}}_{\text{th}}), \quad (6.25)$$

where $\Theta(x)$ is the Heaviside step function.

6.2.4. Mismatch and Template Banks

In wide-parameter-space searches the unknown signal parameters $\lambda \in \mathbb{P}$ are assumed to fall somewhere within a given search space \mathbb{P} . In this case one needs to compute a statistic (such as those defined in the previous sections) over a whole “bank” of templates $\mathbb{T} \equiv \{\lambda_i\}_{i=1}^N$. This template bank has to be chosen in such a way that any putative signal $\lambda_s \in \mathbb{P}$ would suffer only an acceptable level of loss of SNR. This can be quantified in terms of the so-called *mismatch* μ , defined as the relative loss of $\rho^2(\lambda_s; \lambda)$ in a template λ with respect to the perfect-match $\rho^2(\lambda_s; \lambda_s) = \rho_0^2$ (of Eq. (6.15)), namely

$$\mu(\lambda_s; \lambda) \equiv \frac{\rho^2(\lambda_s; \lambda_s) - \rho^2(\lambda_s; \lambda)}{\rho^2(\lambda_s; \lambda_s)}. \quad (6.26)$$

We can therefore express the “effective” non-centrality parameter ρ_{eff}^2 in a template point λ in the \mathcal{F} -statistic χ^2 -distribution of Eqs. (6.13),(6.22) as

$$\rho_{\text{eff}}^2 \equiv \rho^2(\lambda_s; \lambda) = (1 - \mu) \rho_0^2. \quad (6.27)$$

6.2.5. Sensitivity Depth

The \mathcal{F} -statistic non-centrality parameter ρ^2 depends on signal amplitude h_0 and overall noise floor \mathcal{S} (cf. Eq. (6.20)) only through the combination $h_0/\sqrt{\mathcal{S}}$, as seen in Eq. (6.15). The sensitivity of a search is therefore most naturally characterized in terms of the so-called *sensitivity depth* [199], defined as

$$\mathcal{D} \equiv \frac{\sqrt{\mathcal{S}}}{h_0}, \quad (6.28)$$

in terms of the overall noise PSD \mathcal{S} defined as the harmonic mean over all SFTs used in the search, see Eq. (6.20).

A particular choice of search parameters ($N_{\text{seg}}, T_{\text{data}}$, template bank) in general yields a frequency-dependent upper limit $h_0(f)$, due to the frequency-dependent noise floor $\mathcal{S}(f)$. However, for fixed search parameters this will correspond to a *constant* sensitivity depth \mathcal{D} , which is therefore often a more practical and natural way to characterize the performance of a search, independently of the noise floor.

6.3. Sensitivity Estimate

As discussed in more detail in the introduction, by *sensitivity* we mean the (measured or expected) upper limit on h_0 for a given search (or equivalently, the sensitivity depth $\mathcal{D} = \sqrt{\mathcal{S}}/h_0$), which can either refer to the frequentist or Bayesian upper limit.

6.3.1. Frequentist Upper Limits

The frequentist upper limit is defined as the weakest signal amplitude h_0 that can be detected at a given detection probability p_{det} ³ (typically chosen as 90% or 95%) above a threshold d_{th} on a statistic $d(x)$. The threshold can be chosen as the loudest candidate obtained in the search, or it can be set corresponding to a desired false-alarm level p_{fa} (or p-value), defined as

$$p_{\text{fa}}(d_{\text{th}}) \equiv P(d > d_{\text{th}} \mid h_0 = 0), \quad (6.29)$$

which can be inverted to yield $d_{\text{th}} = d_{\text{th}}(p_{\text{fa}})$. The detection probability for signals of amplitude h_0 is

$$p_{\text{det}}(d_{\text{th}}; h_0) \equiv P(d > d_{\text{th}} \mid h_0), \quad (6.30)$$

which can be inverted to yield the upper limit $h_0(d_{\text{th}}, p_{\text{det}})$.

We can write $p_{\text{fa}}(d_{\text{th}}) = p_{\text{det}}(d_{\text{th}}; h_0 = 0)$, and so we can express both in terms of the general threshold-crossing probability as

$$P(d > d_{\text{th}} \mid h_0) = \int_{d_{\text{th}}}^{\infty} P(d \mid h_0) dd. \quad (6.31)$$

6.3.2. Approximating Wide-parameter-space Statistics

As discussed in Sec. 6.2.4, a wide parameter-space search for unknown signals $\lambda \in \mathbb{P}$ normally proceeds by computing a (single-template) statistic over a bank of templates $\mathbb{T} \equiv \{\lambda_i\}_{i=1}^N$ covering the parameter space \mathbb{P} . This results in a corresponding set of (single-template) statistic

³or equivalently, false-dismissal probability $p_{\text{fd}} = 1 - p_{\text{det}}$

values $\{d^1(x; \lambda_i)\}$, which need to be combined to form the overall wide-parameter-space statistic $d(x)$. This would naturally be obtained via marginalization (i.e. integrating the likelihood over \mathbb{P}), but in practice is mostly done by maximizing the single-template statistic over \mathbb{T} , i.e.

$$d(x) \equiv d^*(x) \equiv \max_{\lambda_i \in \mathbb{T}} d^1(x; \lambda_i). \quad (6.32)$$

Noise Case

It is difficult to determine a reliable expression for $P(d^* | h_0 = 0)$ for the pure noise case of Eq. (6.29), even if the single-template statistic $P(d^1 | h_0 = 0)$ follows a known distribution (such as for the \mathcal{F} -based statistics discussed in Sec. 6.2). The reason for this difficulty lies in the correlations that generally exist between “nearby” templates in $\lambda_i \in \mathbb{T}$.

If all \mathcal{N} templates were strictly uncorrelated, one could use the well-known expression Eq. (6.73) [29, 200] for the distribution of the maximum. In this case one can also relate the single-trial p-value $p_{\text{fa}}^1 \approx p_{\text{fa}}/\mathcal{N}$ to the wide-parameter-space p-value p_{fa} (for $p_{\text{fa}}^1 \ll 1$).

Although it is a common assumption in the literature, template correlations do not simply modify the “effective” number of independent templates to use in Eq. (6.73), but they generally also affect the functional form of the underlying distribution for the maximum d^* , as illustrated in appendix 6.11 with a simple toy model.

In this work we assume that the upper limit refers to a known detection threshold in Eq. (6.30). This can be obtained either from (i) the loudest observed candidate (the most common situation in real searches), or from (ii) setting a single-template p-value p_{fa}^1 and inverting the known single-template distribution Eq. (6.29), or from (iii) a numerically-obtained relation between p_{fa} and the threshold d_{th} , e.g. via Monte-Carlo simulation.

Signal Case

We assume that the highest value of d^1 will be realized near the signal location, i.e.

$$d^*(x) \approx d^1(x; \lambda^*), \quad (6.33)$$

where λ^* is the “closest” template $\in \mathbb{T}$ to the signal location λ_s , defined in terms of the mismatch Eq. (6.26). This template yields the highest effective non-centrality parameter over the template bank, namely

$$\rho_{\text{eff}}^2 \equiv \rho^2(\lambda_s; \lambda^*) = (1 - \mu) \rho_0^2(\lambda_s). \quad (6.34)$$

This assumption turns out to be valid as long as the p-value p_{fa} is low (typically $p_{\text{fa}} \lesssim 1\%$) and the signals have relatively high detection probability (typically $p_{\text{det}} \sim 90\%$ or 95%). However, in Sec. 6.5 we also encounter deviations from the predictions that can be traced to violations of this assumption.

6.3.3. StackSlide- \mathcal{F} Sensitivity

We first consider a semi-coherent StackSlide- \mathcal{F} search using the summed $\hat{\mathcal{F}}$ -statistic of Eq. (6.21), i.e. $d^1(x; \lambda) = 2\hat{\mathcal{F}}(x; \lambda)$. This case also includes fully-coherent \mathcal{F} -statistic searches, which simply correspond to the special case $N_{\text{seg}} = 1$.

We see from Eq. (6.31) that in order to estimate the sensitivity, we need to know $P(2\hat{\mathcal{F}} | h_0)$. This can be obtained via marginalization (at fixed h_0) of the known distribution $P(2\hat{\mathcal{F}} | \rho^2)$ of Eq. (6.22), combined with the assumption Eq. (6.34) that the highest statistic value will occur

in the “closest” template, with mismatch distribution $P(\mu)$:

$$\begin{aligned} P(2\hat{\mathcal{F}} | h_0) &= \int P(2\hat{\mathcal{F}}, \theta, \mu | h_0) d^4\theta d\mu \\ &= \int P(2\hat{\mathcal{F}} | h_0, \theta, \mu) P(\theta) P(\mu) d^4\theta d\mu \\ &= \int P(2\hat{\mathcal{F}} | \rho_{\text{eff}}^2) P(\theta) P(\mu) d^4\theta d\mu, \end{aligned} \quad (6.35)$$

where $\rho_{\text{eff}}^2(h_0, \theta, \mu) = \rho_0^2(h_0, \theta) (1 - \mu)$ in terms of the perfect-match non-centrality ρ_0^2 defined in Eq. (6.15), and in the last step we used the fact that the distribution for $2\hat{\mathcal{F}}$ is fully specified in terms of the non-centrality parameter ρ^2 of the χ^2 -distribution with $4N_{\text{seg}}$ degrees of freedom, as given in Eq. (6.22).

Equation (6.35) requires five-dimensional integration for each sensitivity estimation, which would be slow and cumbersome. One of the key insights in [29] was to notice that the perfect-match SNR ρ_0 of Eq. (6.15) depends on the four parameters θ only through the scalar $R^2(\theta)$, and we can therefore use a reparametrization

$$\int_{\theta(R^2)} P(\theta) d^4\theta = P(R^2) dR^2, \quad (6.36)$$

where $\theta(R^2)$ denotes the subspace of θ values yielding a particular R^2 from Eq. (6.67).

The one-dimensional distribution $P(R^2)$ can be obtained by Monte-Carlo sampling over the priors of sky-position \hat{n} (typically either isotropically over the whole sky, or a single sky-position in case of a directed search) and polarization angles $\cos \iota$ (uniform in $[-1, 1]$) and ψ (uniform in $[-\pi/4, \pi/4]$). The resulting values of $R^2(\theta)$ are histogrammed and used as an approximation for $P(R^2)$, which can be reused for repeated sensitivity estimations with the same θ -priors. We can thus rewrite Eq. (6.35) as

$$P(2\hat{\mathcal{F}} | h_0) = \int P(2\hat{\mathcal{F}} | \rho_{\text{eff}}^2) P(R^2) P(\mu) dR^2 d\mu, \quad (6.37)$$

with

$$P(2\hat{\mathcal{F}} | \rho_{\text{eff}}^2) = \chi_{4N_{\text{seg}}}^2(2\hat{\mathcal{F}}; \rho_{\text{eff}}^2), \quad (6.38)$$

$$\rho_{\text{eff}}^2(h_0, R^2, \mu) = \frac{4}{25} \frac{h_0^2}{\mathcal{S}} T_{\text{data}} R^2 (1 - \mu). \quad (6.39)$$

The mismatch distribution $P(\mu)$ for any given search can be obtained via injection-recovery Monte-Carlo simulation, where signals are repeatedly generated (without noise) and searched for over the template bank, obtaining the corresponding mismatch μ for each injection. This is a common step in validating a search and template-bank setup. Alternatively, for some search methods pre-computed estimates for the mismatch distributions exist as a function of the template-bank parameters, e.g. for the WEAVE search code [145].

Inserting Eq. (6.37) into the detection probability of Eq. (6.31), we obtain

$$p_{\text{det}}(2\hat{\mathcal{F}}_{\text{th}}; h_0) = \int p_{\text{det}}(2\hat{\mathcal{F}}_{\text{th}}; \rho_{\text{eff}}^2) P(R^2) P(\mu) dR^2 d\mu, \quad (6.40)$$

where

$$p_{\text{det}}(2\hat{\mathcal{F}}_{\text{th}}; \rho_{\text{eff}}^2) \equiv \int_{2\hat{\mathcal{F}}_{\text{th}}}^{\infty} \chi_{4N_{\text{seg}}}^2(2\hat{\mathcal{F}}; \rho_{\text{eff}}^2) d2\hat{\mathcal{F}}. \quad (6.41)$$

Equation (6.40) can be easily and efficiently computed numerically, and simple inversion (via 1-D root-finding) yields the sensitivity (i.e. upper limit) h_0 for given detection probability p_{det} and threshold $2\hat{\mathcal{F}}_{\text{th}}$.

6.3.4. Multi-stage StackSlide- \mathcal{F} Sensitivity

The sensitivity estimate for a single StackSlide- \mathcal{F} search can be generalized to hierarchical multi-stage searches, where threshold-crossing candidates of one search stage are followed up by deeper subsequent searches in order to increase the overall sensitivity (e.g. see [126, 129, 139, 188, 201]). We denote the n stages with an index $i = 1 \dots n$. Each stage i is characterized by the number $N_{\text{seg}}^{(i)}$ of segments, the amount of data $T_{\text{data}}^{(i)}$, the noise PSD $\mathcal{S}^{(i)}$, a mismatch distribution $P(\mu^{(i)})$, and a threshold $2\hat{\mathcal{F}}_{\text{th}}^{(i)}$ (corresponding to a false-alarm level $p_{\text{fa}}^{(i)}$ at that stage).

The initial wide-parameter-space search (stage $i = 1$) yields candidates that cross the threshold $2\mathcal{F}_{\text{th}}^{(1)}$ in certain templates $\{\lambda\}$. The next stage follows up these candidates with a more sensitive search, which can be achieved by reducing the mismatch $\mu^{(i)}$ (choosing a finer template bank grid), or by increasing the coherent segment length (and reducing the number of segments $N_{\text{seg}}^{(i)}$). Often the final stage $i = n$ in such a follow-up hierarchy would be fully coherent, i.e. $N_{\text{seg}}^{(n)} = 1$.

In order for any given candidate (which can be either due to noise or a signal) to cross the final threshold $2\mathcal{F}^{(n)}$, it has to cross *all* previous thresholds as well, in other words Eqs. (6.29) and (6.30) now generalize to

$$p_{\text{det}}^{(\text{tot})}(h_0) = P(\{2\hat{\mathcal{F}}^{(i)} > 2\hat{\mathcal{F}}_{\text{th}}^{(i)}\}_{i=1}^n | h_0). \quad (6.42)$$

In order to make progress at this point we need to assume that the threshold-crossing probabilities in different stages are *independent* of each other, so for $j \neq i$ we assume

$$P(2\hat{\mathcal{F}}^{(i)} > 2\hat{\mathcal{F}}_{\text{th}}^{(i)} | \rho^2, 2\hat{\mathcal{F}}^{(j)} > 2\hat{\mathcal{F}}_{\text{th}}^{(j)}) = P(2\hat{\mathcal{F}}^{(i)} > 2\hat{\mathcal{F}}_{\text{th}}^{(i)} | \rho^2), \quad (6.43)$$

which would be exactly true if the different stages used different data (see also [129]). In the case where the same data is used in different stages, this approximation corresponds to an *uninformative* approach, in the sense that we do not know how to quantify and take into account the correlations between the statistics in different stages. We proceed without using this potential information, which could in principle be used to improve the estimate. It is not clear if and how much of an overall bias this approximation would introduce. A detailed study of this question is beyond the scope of this work and will be left for future study.

Using the assumption of independent stages we write

$$p_{\text{det}}^{(\text{tot})}(h_0) = \int \prod_{i=1}^n p_{\text{det}}^{(i)}(2\hat{\mathcal{F}}_{\text{th}}^{(i)}; h_0, R^2) P(R^2) dR^2, \quad (6.44)$$

$$p_{\text{fa}}^{(\text{tot})} = \prod_{i=1}^n p_{\text{fa}}^{(i)}(2\hat{\mathcal{F}}_{\text{th}}^{(i)}), \quad (6.45)$$

where now the R^2 -marginalization needs to happen over all stages combined, as the signal parameters $R^2(\theta)$ are intrinsic to the signal and hence independent of the stage. On the other hand, the mismatch distribution $P(\mu^{(i)})$ depends on the stage, as each stage will in general use a different template grid, and so we have

$$p_{\text{det}}^{(i)}(2\hat{\mathcal{F}}_{\text{th}}^{(i)}; h_0, R^2) = \int_0^1 p_{\text{det}}^{(i)}(2\hat{\mathcal{F}}_{\text{th}}^{(i)}; \rho_{\text{eff}}^2)^{2(i)} P(\mu^{(i)}) d\mu^{(i)}, \quad (6.46)$$

where $p_{\text{det}}(2\hat{\mathcal{F}}_{\text{th}}; \rho_{\text{eff}}^2)$ is given by Eq. (6.41) using the respective per-stage values.

Equation (6.44) can easily be solved numerically and inverted for the sensitivity h_0 at given $p_{\text{det}}^{(\text{tot})}$ and a set of thresholds $\{2\hat{\mathcal{F}}_{\text{th}}^{(i)}\}$.

Note that in practice (e.g. [201]) one would want to choose the thresholds in such a way that a signal that passed the 1st-stage threshold $2\hat{\mathcal{F}}_{\text{th}}^{(1)}$ should have a very low probability of being discarded by subsequent stages, in other words $p_{\text{det}}^{(i>1)} \approx 1$, and therefore $p_{\text{det}}^{(\text{tot})}(h_0) \approx p_{\text{det}}^{(1)}(2\hat{\mathcal{F}}_{\text{th}}^{(1)}; h_0)$. Therefore, subsequent stages mostly serve to reduce the total false-alarm level $p_{\text{fa}}^{(\text{tot})}$, allowing one to increase the first-stage $p_{\text{fa}}^{(1)}$ by lowering the corresponding threshold $\hat{\mathcal{F}}^{(1)}$, resulting in an overall increased sensitivity.

6.3.5. Hough- \mathcal{F} Sensitivity

Here we apply the sensitivity-estimation framework to the Hough- \mathcal{F} statistic introduced in Sec. 6.2.3. We define the per-segment threshold-crossing probability as

$$\begin{aligned} p_{\text{th}}^\ell(h_0, R^2) &\equiv P(2\tilde{\mathcal{F}}_\ell > 2\tilde{\mathcal{F}}_{\text{th}} \mid h_0, R^2) \\ &= p_{\text{det}}^\ell(2\tilde{\mathcal{F}}_{\text{th}}; h_0, R^2) \\ &= \int_0^1 p_{\text{det}}(2\tilde{\mathcal{F}}_{\text{th}}; \rho_{\text{eff},\ell}^2) P(\tilde{\mu}) d\tilde{\mu}, \end{aligned} \quad (6.47)$$

where the per-segment effective SNR $\rho_{\text{eff},\ell}$ is given by replacing T_{data} and \mathcal{S} in Eq. (6.39) with the per-segment quantities T_{data}^ℓ and \mathcal{S}^ℓ . For the per-segment mismatch $\tilde{\mu}^\ell$ we assume that the distribution $P(\tilde{\mu})$ is the same for all segments.

The key approximation for the estimate is that for a given signal $\{h_0, R^2(\theta)\}$, the coherent per-segment $\tilde{\mathcal{F}}_\ell$ -statistic has the same threshold-crossing probability p_{th} in every segment ℓ , i.e. $p_{\text{th}}^\ell = p_{\text{th}}$ for all $\ell = 1 \dots N_{\text{seg}}$. This allows us to write the probability for the Hough number count n_c of Eq. (6.25) for a fixed signal $\{h_0, R^2\}$ as a binomial distribution:

$$P(n_c \mid h_0, R^2) = \binom{N_{\text{seg}}}{n_c} p_{\text{th}}^{n_c} (1 - p_{\text{th}})^{N_{\text{seg}} - n_c}, \quad (6.48)$$

with $p_{\text{th}}(h_0, R^2)$ given by Eq. (6.47). For a given threshold $n_{c,\text{th}}$ on the number count the detection probability is

$$p_{\text{det}}(n_{c,\text{th}}; h_0, R^2) = \sum_{n_c=n_{c,\text{th}}}^{N_{\text{seg}}} P(n_c \mid h_0, R^2). \quad (6.49)$$

Marginalization over R^2 yields the detection probability at fixed amplitude h_0 :

$$p_{\text{det}}(n_{c,\text{th}}; h_0) = \int p_{\text{det}}(n_{c,\text{th}}; h_0, R^2) P(R^2) dR^2. \quad (6.50)$$

We can numerically solve this for h_0 at given p_{det} and number-count threshold $n_{c,\text{th}}$ yielding the desired sensitivity estimate.

6.3.6. Bayesian Upper Limits

Bayesian upper limits are conceptually quite different [202] from the frequentist ones discussed up to this point. A Bayesian upper limit h_0^C of given confidence (or ‘‘credible level’’) C corresponds to the interval $[0, h_0^C]$ that contains the true value of h_0 with probability C . We can compute this from the posterior distribution $P(h_0 \mid x)$ for the signal-amplitude h_0 given data x , namely

$$C = P(h_0 < h_0^C \mid x) = \int_0^{h_0^C} P(h_0 \mid x) dh_0. \quad (6.51)$$

The Bayesian targeted searches (here referred to as *BayesPE*) for known pulsars (see Table 6.5 and Sec. 6.8.5) compute the posterior $P(h_0 \mid x)$ directly from the data x , using a time-domain method introduced in [112].

Here we focus instead on \mathcal{F} -statistic-based searches over a template bank. As discussed in [202], to a very good approximation we can compute the posterior from the loudest candidate $2\mathcal{F}^*(x)$ found in such a search, using this as a proxy for the data x , i.e.

$$P(h_0 \mid x) \approx P(h_0 \mid 2\mathcal{F}^*(x)) \quad (6.52)$$

$$\propto P(2\mathcal{F}^*(x) \mid h_0) P(h_0), \quad (6.53)$$

where we used Bayes’ theorem. The proportionality constant is determined by the normalization condition $\int P(h_0 | x) dh_0 = 1$.

We have already derived the expression for $P(2\mathcal{F} | h_0)$ in Eq. (6.37). Hence for any choice of prior $P(h_0)$ we can now compute the Bayesian upper limit $h_0^C(2\mathcal{F}^*)$ for given loudest candidate $2\mathcal{F}^*$ by inverting Eq. (6.51).

It is common for Bayesian upper limits on the amplitude to choose a uniform prior in h_0 (e.g. see [203]), which has the benefit of simplicity, and also puts relatively more weight on larger values of h_0 than might be physically expected (weaker signals should be more likely than stronger ones). This prior therefore results in larger, i.e. “more conservative”, upper limits than a more physical prior would.

Note that the Bayesian ULs of targeted searches for known pulsars (see Sec. 6.8.5) compute the h_0 -posterior directly from the data rather than from an \mathcal{F} -statistic. Therefore we cannot use a known threshold or loudest candidate $2\mathcal{F}^*$ for inverting Eq. (6.51) and hence we cannot apply the above framework directly. We instead compute an *expected depth* by calculating estimates for $2\mathcal{F}^*$ -values drawn randomly from the central χ_4^2 -distribution and averaging the results.

6.3.7. Numerical Implementation

The expressions for the various different sensitivity estimates of the previous sections have been implemented in GNU OCTAVE [204], and are available as part of the `OctApps` [205] data-analysis package for continuous gravitational waves.

The function to estimate (and cache for later reuse) the distribution $P(R^2)$ of Eq. (6.36) is implemented in `SqrSNRGeometricFactorHist()`. The sensitivity-depth estimate for `StackSlide- \mathcal{F}` -searches is implemented in `SensitivityDepthStackSlide()`, both for the single-stage case of Eq. (6.40) and for the general multi-stage case of Eq. (6.44). For single-stage `StackSlide- \mathcal{F}` there is also a function `DetectionProbabilityStackSlide()` estimating the detection probability for a given signal depth \mathcal{D} and detection threshold.

The Hough- \mathcal{F} sensitivity estimate of Eq. (6.50) is implemented in `SensitivityDepthHoughF()`. An earlier version of this function had been used for the theoretical sensitivity comparison in [196] (Sec. VB, and also [206]), where it was found to agree within an rms error of 7% with the measured upper limits. The Bayesian \mathcal{F} -based upper limit expression Eq. (6.51) is implemented in `SensitivityDepthBayesian()`.

Typical input parameters are the number of segments N_{seg} , the total amount of data T_{data} , the mismatch distribution $P(\mu)$, name of detectors used, single-template false-alarm level p_{fa}^1 (or alternatively, the \mathcal{F} -statistic threshold), and the confidence level p_{det} . The default prior on sky-position is isotropic (suitable for an all-sky search), but this can be restricted to any sky-region (suitable for directed or targeted searches).

The typical runtime on a ThinkPad P51 with 3GHz Intel Xeon E3 for a sensitivity estimate including computing $P(R^2)$ (which is the most expensive part) is about 25 seconds per detector. When reusing the same θ -prior on subsequent calls, a cached $P(R^2)$ is used and the runtime is reduced to about 10 seconds total, independently of the number of detectors used.

6.4. Determining Frequentist Upper Limits

In order to determine the frequentist upper limit (UL) on the signal amplitude h_0 defined in Eq. (6.30), one needs to quantify the probability that a putative signal with fixed amplitude h_0 (and all other signal parameters drawn randomly from their priors) would produce a statistic value exceeding the threshold (corresponding to a certain false-alarm level, or p-value). The upper limit on h_0 is then defined as the value $h_0^{p_{\text{det}}}$ for which the detection probability is exactly p_{det} , typically chosen as 90% or 95%, which is often referred to as the *confidence level* of the UL.

Note that here and in the following it will often be convenient to use the sensitivity depth $\mathcal{D} \equiv \sqrt{\mathcal{S}}/h_0$ introduced in Sec. 6.2.5 instead of the amplitude h_0 . We denote $\mathcal{D}^{p_{\text{det}}}$ as the sensitivity depth corresponding to the upper limit $h_0^{p_{\text{det}}}$ (note that this corresponds to a *lower limit* on depth).

The UL procedure is often implemented via a Monte-Carlo injection-and-recovery method: A signal of fixed amplitude $h_0 = \sqrt{\mathcal{S}}/\mathcal{D}$ and randomly-drawn remaining parameters is generated in software and added to the data (either to real detector data or to simulated Gaussian noise). This step is referred to as a *signal injection*. A search is then performed on this data, and the loudest statistic value \mathcal{F}^* is recorded and compared against the detection threshold \mathcal{F}_{th} . Repeating this injection and recovery step many times and recording the fraction of times the threshold is exceeded yields an approximation for $p_{\text{det}}(\mathcal{F}_{\text{th}}; \mathcal{D})$. By repeating this procedure over different \mathcal{D} values and interpolating one can find $\mathcal{D}^{p_{\text{det}}}$ corresponding to the desired detection probability (and therefore also $h_0^{p_{\text{det}}}$).

We distinguish in the following between *measured* and *simulated* upper limits:

- *Measured ULs* refer to the published UL results obtained on real detector data. These typically use an identical search procedure for the ULs as in the actual search, often using the loudest candidate (over some range of the parameter space) from the original search as the corresponding detection threshold for setting the UL. The injections are done in real detector data, and normally the various vetoes, data-cleaning and follow-up procedures of the original search are also applied in the UL procedure.
- *Simulated ULs* are used in this work to verify the accuracy of the sensitivity estimates. They are obtained using injections in simulated Gaussian noise, and searching only a small box in parameter space around the injected signal locations. The box size is empirically determined to ensure that the loudest signal candidates are always recovered within the box. Only the original search statistic is used in the search without any further vetoes or cleaning.

A key difference between (most) published (measured) ULs and our simulated ULs concerns the method of interpolation used to obtain $\mathcal{D}^{p_{\text{det}}}$: In practice this is often obtained via a sigmoid p_{det} -interpolation approach (Sec. 6.4.1), while we use (and advocate for) a (piecewise) linear threshold interpolation (Sec. 6.4.1) instead.

6.4.1. Sigmoid p_{det} Interpolation

In this approach one fixes the detection threshold \mathcal{F}_{th} and determines the corresponding p_{det} for any given fixed- \mathcal{D} injection set. The corresponding functional form of $p_{\text{det}}(\mathcal{D})$ has a qualitative “sigmoid” shape as illustrated in Fig. 6.1. An actual sigmoid function of the form

$$y(\mathcal{D}) = \frac{1}{1 + e^{-k(\mathcal{D}-\mathcal{D}_0)}}, \quad (6.54)$$

is then fitted to the data by adjusting the free parameters k and \mathcal{D}_0 , and from this one can obtain an interpolation value for $\mathcal{D}^{p_{\text{det}}}$.

One problem with this method is that the actual functional form of $p_{\text{det}}(\mathcal{D})$ is not analytically known, and does not actually seem to be well described by the sigmoid of Eq. (6.54), as seen in Fig. 6.1. In this particular example the true value at $p_{\text{det}} = 90\%$ just so happens to lie very close to the sigmoid fit, but the deviation is quite noticeable at $p_{\text{det}} = 95\%$ (see the zoomed inset in Fig. 6.1).

Another problem with this method is that the range of depths required to sample the relation $p_{\text{det}}(\mathcal{D})$ often needs to be quite wide, due to initial uncertainties about where the UL value would be found, which can compound the above-mentioned sigmoid-fitting problem. Furthermore, the

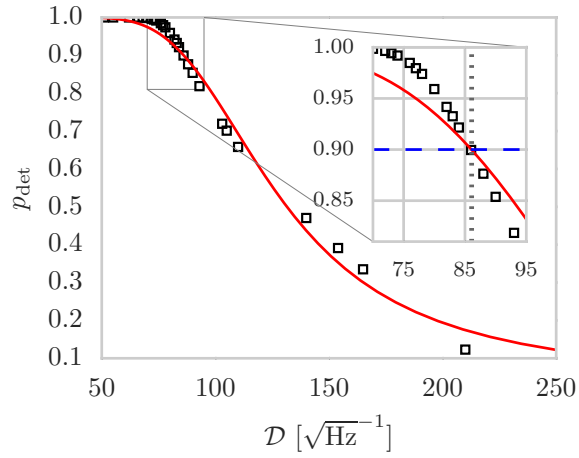


Figure 6.1.: Detection probability p_{det} versus sensitivity depth \mathcal{D} for the **S6-CasA-StackSlide- \mathcal{F}** search (cf. Table 6.2 and Sec. 6.8.3), using a detection threshold of $2\overline{\mathcal{F}}_{\text{th}} = 8$. The squares indicate the results from a simulation in Gaussian noise, while the solid line gives the best-fit sigmoid of Eq. (6.54).

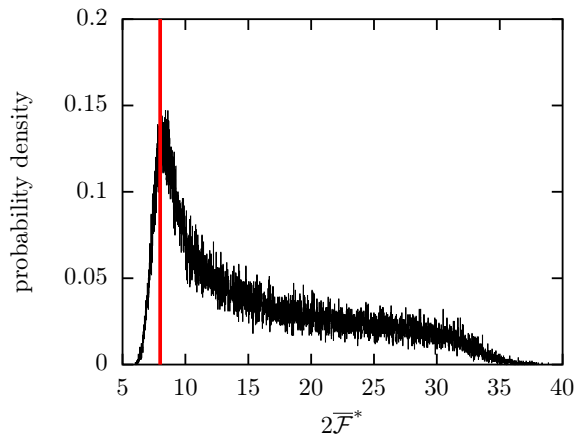


Figure 6.2.: Histogram of recovered loudest $2\overline{\mathcal{F}}$ values for repeated searches on signal injections at fixed sensitivity depth $\mathcal{D} = 86 \text{ Hz}^{-1/2}$ (with all other signal parameters randomized), using the search setup of the **S6-CasA-StackSlide- \mathcal{F}** directed search. The vertical line indicates the resulting threshold value $2\overline{\mathcal{F}}_{\text{th}} = 7.995$ corresponding to $p_{\text{det}} = 90\%$ for this injection set.

injection-recovery step can be quite computationally expensive, limiting the number of trials and further increasing the statistical uncertainty on the p_{det} measurements.

Both of these problems can be mitigated to some extent by using the sensitivity-estimation method described in Sec. 6.3 to obtain a fairly accurate initial guess about the expected UL value, and then sample only in a small region around this estimate, in which case even a linear fit would probably yield good accuracy.

6.4.2. Piecewise-linear Threshold Interpolation

An alternative approach is used in this work to obtain the simulated ULs: For each set of fixed- \mathcal{D} injections and recoveries, we determine the threshold on the statistic required in order to obtain the desired detection fraction p_{det} . This is illustrated in Fig. 6.2, which shows a histogram of the observed loudest $2\bar{\mathcal{F}}$ candidates obtained in each of $N = 10^4$ injection and recovery runs at a fixed signal depth of $\mathcal{D} = 86 \text{ Hz}^{-1/2}$, using the **S6-CasA-StackSlide- \mathcal{F}** search setup (cf. Sec. 6.8.3). By integrating the probability density from $2\bar{\mathcal{F}} = 0$ until we reach the desired value $1 - p_{\text{det}}$, we find the detection threshold $2\bar{\mathcal{F}}_{\text{th}}$ at this signal depth \mathcal{D} . Repeating this procedure at different depths therefore generates a sampling of the function $\mathcal{D}^{p_{\text{det}}}(2\bar{\mathcal{F}}_{\text{th}})$, illustrated in Fig. 6.3. These points can be interpolated to the required detection threshold, which yields the desired upper-limit depth $\mathcal{D}^{p_{\text{det}}}$.

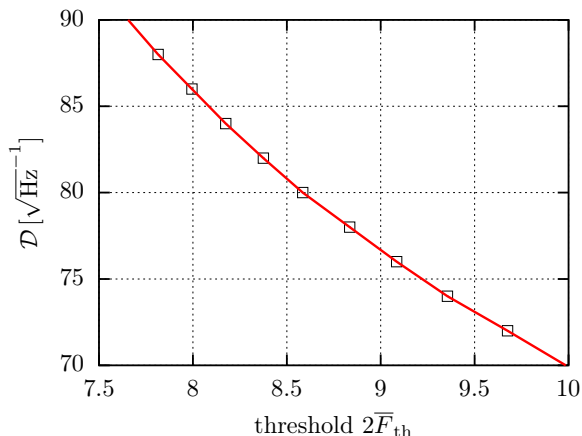


Figure 6.3.: Sensitivity depth versus detection threshold. Boxes and solid lines indicate the piecewise-linear interpolation through the obtained thresholds at different depths of an injection-recovery simulation, using the **S6-CasA-StackSlide- \mathcal{F}** search setup ([185] and Sec. 6.8.3).

We see in in Fig. 6.3 that this function appears to be less “curvy” in the region of interest compared to $p_{\text{det}}(\mathcal{D})$ shown in Fig. 6.1. This allows for easier fitting and interpolation, for example a linear or quadratic fit should work quite well. In fact, here we have simply used piecewise-linear interpolation, which is sufficient given our relatively fine sampling of signal depths.

As already mentioned in the previous section, using the sensitivity estimate of Sec. 6.3 one can determine the most relevant region of interest beforehand and focus the Monte-Carlo injection-recoveries on this region, which will help ensure that any simple interpolation method will work well.

Alternatively, for either the $p_{\text{det}}(\mathcal{D})$ - or the $\mathcal{D}(2\bar{\mathcal{F}}_{\text{th}})$ -sampling approach, one could also use an iterative root-finding method to approach the desired p_{det} or $2\bar{\mathcal{F}}_{\text{th}}$, respectively.

6.5. Comparing Estimates Against Simulated Upper Limits

In this section we compare the sensitivity estimates from Sec. 6.3 against simulated ULs for two example cases (an all-sky search and a directed search), in order to quantify the accuracy and reliability of the estimation method and implementation. This comparison shows generally good agreement, and also some instructive deviations.

Both examples are wide-parameter-space searches using a template bank over the unknown signal parameter dimensions (namely, {sky, frequency and spindown} in the all-sky case, and {frequency and first and second derivatives} in the directed-search case).

The simulated-UL procedure (see Sec. 6.4) performs a template-bank search over a box in parameter space containing the injected signal (at a randomized location) in Gaussian noise. On the other hand, the sensitivity estimate (cf. Eq. (6.40)) uses the mismatch distribution $P(\mu)$ obtained for this template bank via injection-recovery box searches on signals without noise. We refer to this in the following as the *box search*.

It will be instructive to also consider the (unrealistic) case of a perfectly-matched search, using only a single template that matches the signal parameters perfectly for every injection, corresponding to zero mismatch $\mu = 0$ in Eq. (6.40). We refer to this as the *zero-mismatch search*.

6.5.1. Example: S6-AllSky-StackSlide- \mathcal{F} Search

In this example we use the setup of the all-sky search S6-AllSky-StackSlide- \mathcal{F} [207], which was using the GCT implementation [138] of the StackSlide- \mathcal{F} statistic and was performed on the volunteer-computing project Einstein@Home [7], see Table 6.1 and Sec. 6.8.2 for more details.

Figure 6.4 shows the comparison between simulated ULs and estimated sensitivity depths $\mathcal{D}^{90\%}$ versus threshold $2\bar{\mathcal{F}}_{\text{th}}$, for the *box search* (squares and solid line), as well as for the *zero-mismatch search* (crosses and dashed line). We see excellent agreement between estimated and

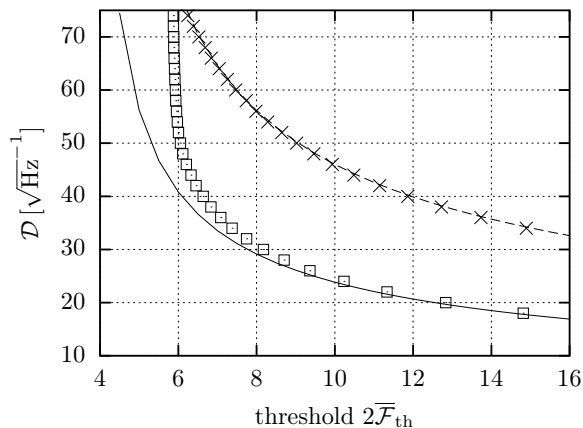


Figure 6.4.: Comparison of estimated and simulated sensitivity depth $\mathcal{D}^{90\%}$ as a function of threshold $2\bar{\mathcal{F}}_{\text{th}}$ for the S6-AllSky-StackSlide- \mathcal{F} search [207]. The solid line shows the UL estimate for the *box search*, and the squares (\square) show the corresponding simulated ULs. The dashed line indicates the estimate for the *zero-mismatch case*, and the crosses (\times) are for the simulated zero-mismatch ULs. In the box search we observe an increasing divergence at decreasing thresholds due to noise effects, discussed in Sec. 6.5.1.

simulated ULs for the zero-mismatch search. We also find very good agreement for the box-search at higher thresholds, while we see an increasing divergence $\mathcal{D} \rightarrow \infty$ of the simulated ULs at decreasing thresholds, which is not captured by the estimate.

This discrepancy can be understood as the effect of noise fluctuations, which can enter in two different ways (that are not completely independent of each other):

- (i) For decreasing thresholds the corresponding false-alarm level Eq. (6.29) grows, as it becomes increasingly likely that a “pure noise” candidate (i.e. unrelated to a signal) crosses the threshold. In the extreme case where p_{fa} approaches p_{det} , the frequentist upper limit would tend to $h_0 \rightarrow 0$, corresponding to $\mathcal{D} \rightarrow \infty^4$. This is illustrated in Fig. 6.5 showing the distribution of the loudest $2\bar{\mathcal{F}}$ in a box search on pure Gaussian noise, which can be compared to the diverging depth of the simulated box search around $2\bar{\mathcal{F}}_{\text{th}} \lesssim 6$ in Fig. 6.4.

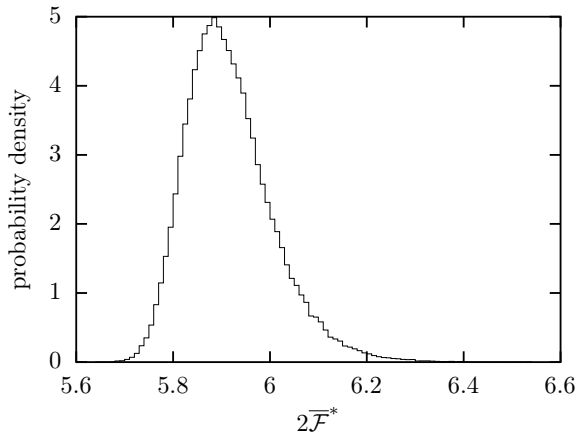


Figure 6.5.: Distribution of the loudest $2\bar{\mathcal{F}}$ for a box search on pure Gaussian noise, using the S6-AllSky-StackSlide- \mathcal{F} search setup.

We note that the procedures used for *measured ULs* in CW searches typically make sure that the detection threshold has a very small false-alarm level, and we thus expect this effect to have a negligible impact in cases of practical interest.

- (ii) The sensitivity estimate for wide-parameter-space searches makes the assumption that the loudest candidate $2\bar{\mathcal{F}}^*$ is always found in the *closest* template to the signal (i.e. with the smallest mismatch μ), as discussed in Sec. 6.3.2. However, while the closest template has the highest *expected* statistic value (by definition), other templates can actually produce the loudest statistic value in any given noise realization. How likely that is to happen depends on the details of the parameter space, the template bank and the threshold. Generally it is more likely at lower thresholds, as more templates further away from the signal are given a chance to cross the threshold (despite their larger mismatch).

The *true* distribution $P(2\bar{\mathcal{F}}^* | h_0)$ of a box search will therefore be shifted to higher values compared to the approximate distribution used in Eq. (6.37). This implies that an actual search can have a higher detection probability than predicted by the estimate (corresponding to a larger sensitivity depth).

Both of these effects contribute to different extents to the box-search discrepancy in Fig. 6.4 at lower thresholds:

The sampling distribution for $2\bar{\mathcal{F}}^*$ in the presence of relatively strong signals at $\mathcal{D} = 20 \text{ Hz}^{-1/2}$ is shown in the left plot of Fig. 6.6, both for a simulated box search as well as for the assumed distribution in the estimate. We see that most of the loudest candidates obtained in the simulation are above $2\bar{\mathcal{F}}^* > 9$, and are therefore extremely unlikely to be due to noise alone, as

⁴Bayesian upper limits do not have this property, e.g. see [202] for more detailed analysis of these different types of upper limits.

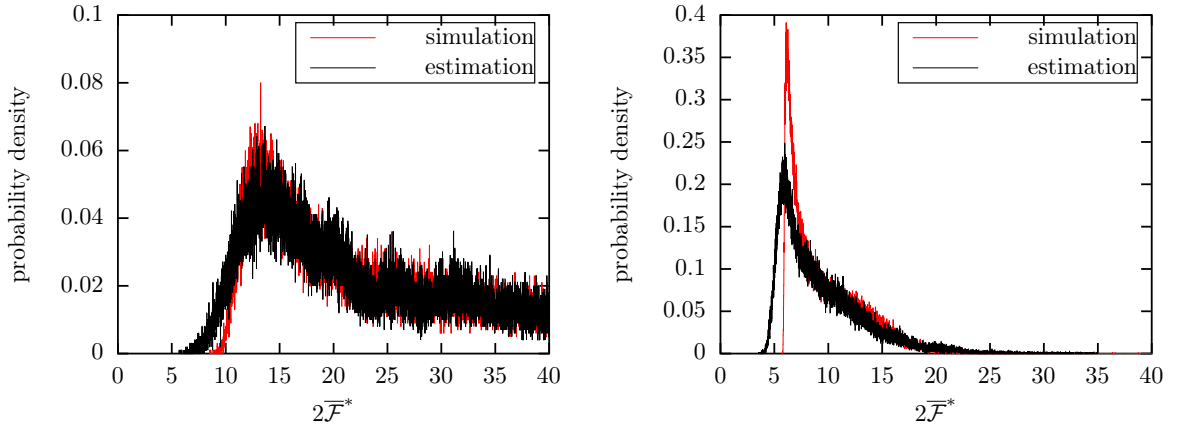


Figure 6.6.: Loudest $2\bar{\mathcal{F}}$ distribution for a box-search (using the `S6-AllSky-StackSlide- \mathcal{F}` setup) with signals at a depth of $\mathcal{D} = 20 \text{ Hz}^{-1/2}$ (left plot) and $\mathcal{D} = 46 \text{ Hz}^{-1/2}$ (right plot). The black histogram shows the assumed distribution for sensitivity *estimation* in Eq. (6.37), and the lighter color shows the histogram obtained in a Monte-Carlo *simulation* with signals injected in Gaussian noise.

seen from Fig. 6.5. The difference between the two distributions in the left plot of Fig. 6.6 is therefore solely due to effect (ii). However, we see in Fig. 6.4 that the resulting discrepancy in the sensitivity estimate at $\mathcal{D} = 20 \text{ Hz}^{-1/2}$ is still very small.

For weaker signals at $\mathcal{D} = 46 \text{ Hz}^{-1/2}$, we see in the right plot of Fig. 6.6 that the corresponding distribution now overlaps with the pure-noise distribution of Fig. 6.5. The sensitivity depth therefore increasingly diverges for thresholds in the range $2\bar{\mathcal{F}}_{\text{th}} \sim [5.8, 6.1]$ due to the increasing impact of effect (i).

6.5.2. Example: Multi-directed 01-MD-StackSlide- \mathcal{F}

In this example we use the search setup of the directed search `01-MD-StackSlide- \mathcal{F}` [191] currently running on Einstein@Home. This search consists of several directed searches for different targets on the sky, including Vela Jr. and Cas-A.

The comparison between simulated and estimated UL depths $\mathcal{D}^{90\%}$ for these two targets is shown in Fig. 6.7. We see again very good agreement (relative deviations $\lesssim 3\%$) in the zero-mismatch case. However, these deviations are larger than in the all-sky case shown in Fig. 6.4. We suspect that this is due to the different antenna-pattern implementations of Eq. (6.65) between the search code and the estimation scripts: We see different signs of the deviation for different sky positions (Vela Jr. versus Cas-A), and the effect disappears when averaging over the whole sky (as seen in Fig. 6.4). However, the small size of the deviations did not warrant further efforts to try to mitigate this.

For the box-search case we see good agreement at higher thresholds, with again increasing deviations at lower thresholds due to the noise effects discussed in the previous all-sky example Sec. 6.5.1.

6.6. Comparing Estimates Against Measured Upper limits

In this section we present a general overview of measured sensitivity depths $\mathcal{D}_{\text{meas}}$ derived from the published upper limits of various past CW searches. For the subset of searches where an \mathcal{F} -statistic-based method was used (and for Bayesian targeted ULs), we provide the sensitivity estimate for comparison.

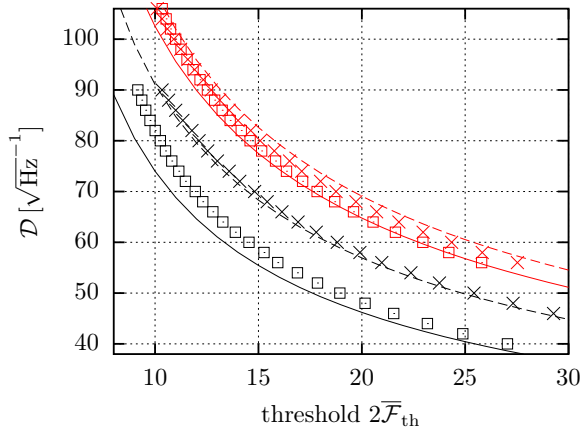


Figure 6.7.: Comparison of estimated and simulated sensitivity depth $\mathcal{D}^{90\%}$ as a function of the threshold $2\bar{\mathcal{F}}_{\text{th}}$ for two targets of the multi-directed search setup 01-MD-StackSlide- \mathcal{F} . The solid lines show the UL estimate for a *box search*, while the squares (\square) show the corresponding simulated ULs. The dashed lines indicate the estimate for the *zero-mismatch case*, and the crosses (\times) are for the simulated zero-mismatch ULs. The upper group of curves are for the target Vela Jr., while the lower group of curves are for Cas A.

The results are summarized in Tables 6.1– 6.4 for the different search categories (all-sky, directed and narrow-band, binary and targeted), and more details about each search are found in Appendix 6.8.

6.6.1. General Remarks and Caveats

Converting Published h_0 ULs into Depths \mathcal{D}

Some searches already provide their upper limits in the form of a sensitivity depth \mathcal{D}^{det} , but in most cases only the amplitude upper-limits h_0^{det} are given. For these latter cases we try to use a reasonable PSD estimate $\mathcal{S}(f)$ for the data used in the search in order to convert the quoted amplitude upper limits into sensitivity depths according to Eq. (6.28). This PSD estimate introduces a systematic uncertainty in the converted depth values, as in most cases we do not have access to the “original” PSD estimate used for the h_0 UL calculation.

In particular, even small differences in windowing or the type of frequency averaging can result in large differences in the PSD estimate near spectral disturbances. This can translate into large differences in the resulting converted depth values. In order to mitigate outliers due to such noise artifacts we quote the *median* over the converted measured depth values $\{\mathcal{D}_k\}$ (where k either runs over multiple frequencies, targets or detectors) and estimate the corresponding standard deviation using the *median absolute deviation* (MAD) [208], namely

$$\begin{aligned} \mathcal{D}^{\text{med}} &\equiv \text{median} [\mathcal{D}_k] , \\ \hat{\sigma} &\equiv 1.4826 \text{ median} \left[\left| \mathcal{D}_k - \mathcal{D}^{\text{med}} \right| \right] . \end{aligned} \quad (6.55)$$

Comparing Different Searches by Sensitivity Depth \mathcal{D}

We can see in the tables 6.1– 6.4 that searches within the same search category often show roughly comparable sensitivity depths. At one end of the spectrum are the fully-targeted searches, for which the parameter space (for each pulsar) is a single point, and one can achieve the maximal possible sensitivity for the available data, namely $\mathcal{D} \sim \mathcal{O}(500 \text{ Hz}^{-1/2})$ (see Table 6.5). At the

other end of the spectrum lies the all-sky binary search with a sensitivity depth of $\mathcal{D} \sim 3 \text{ Hz}^{-1/2}$ (see Table 6.4), which covers the largest parameter space of any search to date.

One cannot directly compare searches on sensitivity depth alone, even within the same search category. Other key aspects of a search are the parameter-space volume covered, the total computing power used, and the robustness of the search to deviations from the assumed signal- or noise-model.

Is it intuitively obvious that the more computing power spent on a fixed parameter-space volume, the more sensitive the search will tend to be, although the increase in sensitivity is typically very weak, often of order the 10th-14th root of the computing power [180].

It is also evident that the larger the parameter space covered by a search, the less sensitivity depth can be achieved due to the increased spending of computing power on “breadth” rather than depth. Ultimately the most directly relevant characteristic of a search would be its *total detection probability* [190, 191], which factors in both breadth and depth as well as the underlying astrophysical prior on signal amplitudes over the parameter space searched.

6.6.2. All-sky Searches

Estimated and measured sensitivity depths for all-sky searches are given in Table 6.1, and further details about individual searches can be found in appendix 6.8.2.

The mean relative error between measured and estimated depths is 9%, while the median error is 7%.

One case of interest is the surprisingly large discrepancy of $\sim 18\%$ observed for the **S6-AllSky-StackSlide- \mathcal{F} +FUP** search, shown in Fig. 6.8, where we see a significantly higher measured depth ($\mathcal{D}_{\text{meas}}^{\text{med}} = 46.9 \text{ Hz}^{-1/2}$) than estimated ($\mathcal{D}_{\text{est}} = 38.3 \text{ Hz}^{-1/2}$).

This can be traced back to the template-maximization approximation used in the estimate, namely effect (ii) discussed in Sec. 6.5.1. The low threshold used in the search ($2\overline{\mathcal{F}}_{\text{th}} = 6.1$) appears to be at the cusp of becoming affected by pure-noise candidates (effect (i) in Sec. 6.5.1), but this effect is still small and does not account for the discrepancy. Furthermore, the upper limit procedure used a multi-stage follow-up, which ensures the final false-alarm level (p-value) is very small, which rules out contamination from pure-noise candidates.

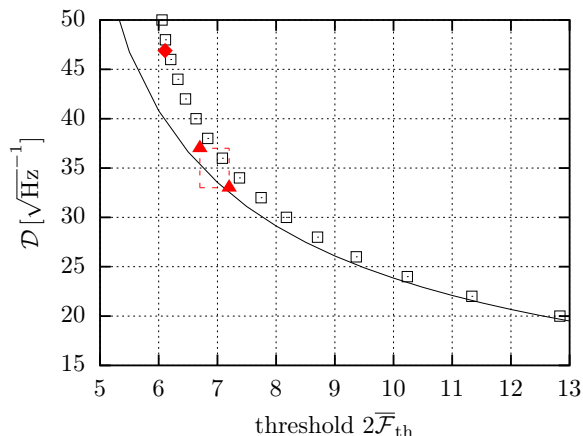


Figure 6.8.: Estimated (—) and simulated (\square) sensitivity depth versus threshold $2\overline{\mathcal{F}}_{\text{th}}$ for the **S6-AllSky-StackSlide- \mathcal{F}** (+FUP) search setup, illustrating the effect of the template-maximization in the estimate (discussed in Sec. 6.5.1). The triangles (Δ) and dashed lines show the measured upper-limit depth $\mathcal{D}_{\text{meas}}^{\text{med}}$ in the initial **S6-AllSky-StackSlide- \mathcal{F}** search [207], and the diamond (\diamond) shows the corresponding result from the follow-up (FUP) search [201] (threshold $2\overline{\mathcal{F}}_{\text{th}} = 6.1$).

Data	Search method	f [Hz]	\dot{f} [$\frac{\text{nHz}}{\text{s}}$]	$\mathcal{D}_{\text{est}}^{\text{est}}$ [$\text{Hz}^{-1/2}$]	$\mathcal{D}_{\text{meas}}^{\text{med}}$ [$\text{Hz}^{-1/2}$]	$\hat{\sigma}_{\mathcal{D}_{\text{meas}}}$ [$\text{Hz}^{-1/2}$]	Ref, Sec
S2	Hough	[200, 400]	[-1.1, 0]	–	11.3	1.5	[209],6.8.2
S2	\mathcal{F}	[160, 728.8]	0	6.5	5.5	1.6	[210],6.8.2
S4	StackSlide	[50, 1000]	[-10, 0]	–	10.5	1.1	[195],6.8.2
S4	Hough	[50, 1000]	[-2.2, 0]	–	13.4	0.7	[195],6.8.2
S4	PowerFlux	[50, 1000]	[-10, 0]	–	{6.1, 21.3} ¹	{0.7, 2.3}	[195],6.8.2
S4	\mathcal{F} +Coinc	[50, 1500]	[-9.5, 1]	–	8.5	0.5	[211],6.8.2
earlyS5	PowerFlux	[50, 1100]	[-5, 0]	–	{16.1, 47.9} ¹	{2.4, 5.9}	[212],6.8.2
earlyS5	\mathcal{F} +Coinc	[50, 1500]	[-12.7, 1.3]	–	10.9	0.2	[213],6.8.2
S5	PowerFlux	[50, 800]	[-6, 0]	–	{25.7, 71.3} ¹	{0.7, 2.2}	[214],6.8.2
S5	Hough- \mathcal{F}	[50, 1190]	[-2, 0.1]	30.5	30.0	1.4	[196],6.8.2
S5	Hough	[50, 1000]	[-0.9, 0]	–	28.1	0.6	[215],6.8.2
S5	StackSlide- \mathcal{F}	[1249.7, 1499.7]	[-2.9, 0.6]	27.0	30.7	–	[216],6.8.2
VSR1	\mathcal{F}_{TD} +Coinc	[100, 1000]	[-16, 0]	–	22.6	6.0	[217],6.8.2
VSR2,4	FreqHough+FUP	[20, 128]	[-0.1, 0.015]	–	35.5	11.1	[218],6.8.2
S6	StackSlide- \mathcal{F}	[50, 510]	[-2.7, 0.3]	34.4	37.0	–	[207],6.8.2
S6	StackSlide- \mathcal{F} +FUP	[50, 510]	[-2.7, 0.3]	38.3	46.9	–	[201],6.8.2
S6	PowerFlux	[100, 1500]	[-11.8, 10]	–	{17.9, 52.8} ¹	{1.4, 3.4}	[219],6.8.2
O1	StackSlide- \mathcal{F}	[20, 100]	[-2.7, 0.3]	46.4	48.7	–	[188],6.8.2
O1	PowerFlux	[20, 200]	[-10, 1]	–	28.9	2.2	[188],6.8.2
O1	PowerFlux	[20, 475]	[-10, 1]	–	{19.9, 54.6} ¹	{1.3, 3.2}	[187],6.8.2
O1	SkyHough	[20, 475]	[-10, 1]	–	22.4	1.1	[187],6.8.2
O1	\mathcal{F}_{TD} +Coinc	[20, 475]	[-10, 1]	–	23.7	2.1	[187],6.8.2
O1	FreqHough	[20, 475]	[-10, 1]	–	21.4	10.6	[187],6.8.2
O1	PowerFlux	[475, 2000]	[-10, 1]	–	{18.6, 50.9} ¹	{1.3, 3.4}	[127],6.8.2
O1	SkyHough	[475, 2000]	[-10, 1]	–	16.8	3.0	[127],6.8.2
O1	\mathcal{F}_{TD} +Coinc	[475, 2000]	[-10, 1]	–	10.9	0.6	[127],6.8.2

Table 6.1.: All-sky searches: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1). The columns labeled f and \dot{f} give the frequency and spindown ranges covered by each search. Sensitivity depths in *italics* refer to 90%-confidence upper limits, while normal font refers to 95%-confidence. See appendix 6.8.2 for further details on the individual results.

¹Sensitivity depths corresponding to worst linear and circular polarization, respectively, cf. Sec. 6.8.1

6.6.3. Directed and Narrow-band Searches

Estimated and measured sensitivity depths for directed and narrow-band searches are given in Tables 6.2 and 6.3, and further details about individual searches can be found in appendix 6.8.3.

The mean relative error between measured and estimated depths is 5 %, and the median error is 1 %.

For the S6-NineYoung- \mathcal{F} search for nine young supernova remnants shown in Table 6.3, the mean relative error between measured and estimated depths is 4 % (median error 4 %).

For two cases of interest we investigated more closely to understand the origin of the observed deviation:

S5-GalacticCenter-StackSlide- \mathcal{F} search [220]: the reason for the relatively large deviation of 19% in this case between $\mathcal{D}_{\text{est}} = 58.2 \text{ Hz}^{-1/2}$ and $\mathcal{D}_{\text{meas}}^{\text{med}} = 72.1 \text{ Hz}^{-1/2}$ can be understood by looking at the details of this search setup: contrary to the assumed uniform averaging of antenna-pattern functions over time (cf. Sec. 6.3.3), this search setup was specifically optimized by choosing the relatively short segments of $T_{\text{seg}} = 11.5$ hours in such a way as to maximize sensitivity, by selecting times of maximal antenna-pattern sensitivity towards the particular sky direction of the galactic center. This is described in more detail in [199], and is quoted there as yielding a sensitivity improvement of about 20%, consistent with the observed enhancement of measured sensitivity compared to our estimate.

S6-CasA-StackSlide- \mathcal{F} search [185]: the deviation between the estimated sensitivity depth $\mathcal{D}_{\text{est}} = 79.6 \text{ Hz}^{-1/2}$ versus the measured value $\mathcal{D}_{\text{meas}}^{\text{med}} = 72.9 \text{ Hz}^{-1/2}$ does not seem very large per se, but is unusual for the estimate because it typically does not tend to *overestimate* sensitivity by that much. A detailed investigation led us to discover a bug in the original upper-limit script used in [185], which resulted in the injection-recovery procedure to sometimes search the wrong box in parameter space, missing the injected signal. By artificially reproducing the bug in our upper limit simulation we are able to confirm that this bug does account for a decrease in detection probability of about 7%, resulting in an underestimation of the upper-limit depth as shown in Fig. 6.9.

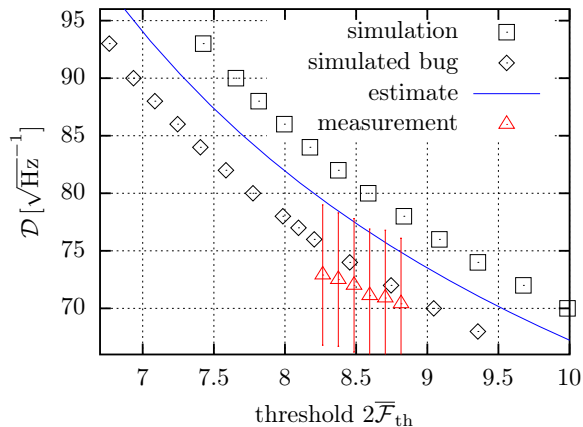


Figure 6.9.: Estimated (—) and simulated (□) sensitivity depth versus threshold $2\overline{\mathcal{F}}_{\text{th}}$ for the S6-CasA-StackSlide- \mathcal{F} search setup [185]. The published upper limits are plotted as triangles (Δ), while the diamonds (\diamond) show the simulated depths if we incorporate the bug found in the original UL procedure.

Science run	Search method	Target	f [Hz]	\mathcal{D}_{est} [$\text{Hz}^{-1/2}$]	$\mathcal{D}_{\text{meas}}^{\text{med}}$ [$\text{Hz}^{-1/2}$]	$\hat{\sigma}_{\mathcal{D}_{\text{meas}}}$ [$\text{Hz}^{-1/2}$]	Ref, Sec
earlyS5	\mathcal{F}	Crab	59.56±0.006	221.3	223.1	–	[221],6.8.3
S5	\mathcal{F}	CasA	[100, 300]	35.9	35.5	0.8	[200],6.8.3
S5	StackSlide- \mathcal{F}	GalacticCenter	[78, 496]	<i>58.2</i>	<i>72.1</i>	4.5	[220],6.8.3
VSR4	5-vector	Vela	22.384±0.02	–	100.5	–	[222],6.8.3
VSR4	5-vector	Crab	59.445±0.02	–	90.1	–	[222],6.8.3
S6	\mathcal{F}	NineYoung (Table 6.3)	[46, 2034]	37.8	37.7	0.3	[184],6.8.3
S6	StackSlide- \mathcal{F}	CasA	[50, 1000]	<i>79.6</i>	<i>72.9</i>	0.4	[185],6.8.3
S6	LooselyCoherent	OrionSpur	[50, 1500]	–	{30.2, 85.7} ²	{2.3, 4.3}	[223],6.8.3
S6	\mathcal{F}	NGC6544	[92.5, 675]	29.3	29.6	1.7	[224],6.8.3
O1	5-vector	11 pulsars	< ±0.1 ¹	–	111.6	12.2	[183],6.8.3
O1	Radiometer	SN1987A	[25, 1726]	–	<i>11.1</i>	4.3	[225],6.8.4
O1	Radiometer	GalacticCenter	[25, 1726]	–	<i>7.7</i>	2.9	[225],6.8.4

Table 6.2.: Directed and narrow-band searches: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1). The column labeled f gives the frequency range covered by each search (omitting \dot{f} and \ddot{f} search ranges). Sensitivity depths in *italics* refer to 90%-confidence upper limits, while normal font refers to 95%-confidence. See appendix 6.8.3 for further details on the individual results.

¹Search band around twice the pulsar spin frequency

²Sensitivity depths corresponding to worst linear and circular polarization, respectively, cf. Sec. 6.8.1

SN remnant Name	G1.9	G18.9	G93.3 DA 530	G111.7 Cas A	G189.1 IC 443	G266.2deep Vela Jr.	G266.2wide Vela Jr.	G291.0 MSH 11-62	G347.3	G350.1
$\mathcal{D}_{\text{est}} [\text{Hz}^{-1/2}]$	29.0	43.9	46.8	29.3	40.1	38.3	24.2	41.1	32.8	37.3
$\mathcal{D}_{\text{meas}}^{\text{med}} [\text{Hz}^{-1/2}]$	28.3	44.4	49.6	31.5	39.2	40.8	26.1	44.0	32.1	36.1
$\hat{\sigma}_{\mathcal{D}_{\text{meas}}} [\text{Hz}^{-1/2}]$	0.8	1.3	1.5	0.9	1.2	1.0	0.7	1.2	0.8	1.1
$T_{\text{data}} [10^6 \text{ s}]$	1.2	3.1	2.8	1.1	2.3	1.9	0.7	2.2	1.4	1.9
$2\mathcal{F}_{\text{th}}$	58.0	56.3	55.6	55.6	55.3	53.7	52.8	56.6	54.1	57.6

Table 6.3.: **S6-NineYoung- \mathcal{F}** search: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1) for nine young supernova remnants [184]. All sensitivity depths refer to 95%-confidence. See appendix 6.8.3 for further details.

Science run	Search method	Target	f [Hz]	$\mathcal{D}_{\text{meas}}^{\text{med}} [\text{Hz}^{-1/2}]$	$\hat{\sigma}_{\mathcal{D}_{\text{meas}}} [\text{Hz}^{-1/2}]$	Ref, Sec
S2	\mathcal{F}	ScoX1	[464, 484],[604, 624]	4.1	0.1	[210],6.8.4
S5	Sideband	ScoX1	[50, 550]	8.1	1.0	[226],6.8.4
S6,VSR2,3	TwoSpect	AllSky	[20, 520]	3.2	0.4	[189],6.8.4
S6,VSR2,3	TwoSpect	ScoX1	[20, 57.25]	8.2	4.0	[189],6.8.4
S6	TwoSpect	ScoX1	[40, 2040]	5.7	1.6	[227],6.8.4
S6	TwoSpect	J1751	{435.5, 621.5, 870.5} ± 1	9.4	1.2	[227],6.8.4
O1	Viterbi	ScoX1	[60, 650]	7.6	1.0	[186],6.8.4
O1	CrossCorr	ScoX1	[25, 2000]	24.0	2.0	[135],6.8.4
O1	Radiometer	ScoX1	[25, 1726]	5.8	1.0	[225],6.8.4

Table 6.4.: Binary searches: measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1). All sensitivity depths refer to 95%-confidence. See appendix 6.8.4 for further details on the individual results.

6.6.4. Searches for Neutron Stars in Binaries

Measured sensitivity depths for searches for CWs from neutron stars in binary systems are given in Table 6.4, and further details about individual searches can be found in appendix 6.8.4. In this case the only \mathcal{F} -statistic-based search is S2-SCO-X1- \mathcal{F} , for which we obtain an estimate of $\mathcal{D}_{\text{est}} = 4.4 \text{ Hz}^{-1/2}$ (assuming an average mismatch of $\mu \sim 0.1/3$ corresponding to a cubic lattice with maximal mismatch of 0.1 [210]). The relative error between measured and estimated sensitivity depth is therefore 8%.

6.6.5. Targeted Searches for Known Pulsars

Estimated and measured sensitivity depths for targeted searches are given in Table 6.5, and further details about individual searches can be found in appendix 6.8.5.

Note that the quoted upper limits of the BayesPE-method are obtained by Bayesian parameter-estimation [112] of $P(h_0 | x)$ directly on the data x . Therefore, we cannot directly apply the Bayesian sensitivity estimate derived in Sec. 6.3.6, which assumes an initial $\mathcal{F}(x)$ -statistic computed on the data, from which the Bayesian upper limit would be derived. We therefore provide an approximate comparison with the *expected* sensitivity estimate, which we compute by estimating depths using $2\mathcal{F}^*$ drawn from a central χ_4^2 distribution (given each target corresponds to a single template) and averaging the resulting estimated \mathcal{D} values. In cases where several targets are covered by the search, we assume for simplicity that the targets are isotropically distributed over the sky and compute a single all-sky sensitivity estimate. For single-target searches the exact sky position is used for the estimate.

The mean relative error between measured and estimated depths is 16%, and the median error is 10%.

Science run	Search method	Targets	$\mathcal{D}_{\text{est}}^{\text{med}}$ [$\text{Hz}^{-1/2}$]	$\hat{\sigma}_{\mathcal{D}_{\text{est}}}$ [$\text{Hz}^{-1/2}$]	$\mathcal{D}_{\text{meas}}^{\text{med}}$ [$\text{Hz}^{-1/2}$]	$\hat{\sigma}_{\mathcal{D}_{\text{meas}}}$ [$\text{Hz}^{-1/2}$]	Ref, Sec
S1	\mathcal{F} (worst-orientation)	J1939+21	70.8	39.8	64.2	38.1	[193],6.8.5
S1	\mathcal{F}	J1939+21	110.4	66.7	101.8	61.8	[193],6.8.5
S1	BayesPE	J1939+21	81.5	19.8	85.2	14.3	[193],6.8.5
S2	BayesPE	28 pulsars	243.5	54.3	156.4	42.2	[228],6.8.5
S3,4	BayesPE	78 pulsars	337.8	81.2	299.5	79.0	[229],6.8.5
earlyS5	BayesPE	Crab	621.3	129.7	774.1	–	[221],6.8.5
S5	BayesPE	116 pulsars	997.8	210.4	932.1	317.1	[230],6.8.5
VSR2	BayesPE, \mathcal{F} ,5-vector	Vela	351.9	78.5	408.5	20.8	[231],6.8.5
S6,VSR2,4	BayesPE, \mathcal{F} ,5-vector	195 pulsars	555.7	116.2	514.7	171.0	[203],6.8.5
O1	BayesPE, \mathcal{F} ,5-vector	200 pulsars	321.6	74.0	355.8	95.4	[182],6.8.5

Table 6.5.: Targeted searches for known pulsars: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (with respectively, median and standard deviation, see Sec. 6.6.1). All sensitivity depths refer to 95%-confidence. See appendix 6.8.5 for further details on the individual results.

6.7. Discussion

In this chapter we presented a fast and accurate sensitivity-estimation framework and implementation for \mathcal{F} -statistic-based search methods for continuous gravitational waves, extending and generalizing an earlier analytic estimate derived by Wette. In particular, the new method is more direct and uses fewer approximations for single-stage StackSlide- \mathcal{F} searches, and is also applicable to multi-stage StackSlide- \mathcal{F} searches, Hough- \mathcal{F} searches and Bayesian upper limits (based on \mathcal{F} -statistic searches).

The typical runtime per sensitivity estimate is about 10 seconds with cached $P(R^2)$ distribution, and about 25 seconds per detector for the first call with a new parameter prior. The accuracy compared to simulated Monte-Carlo upper limits in Gaussian noise is within a few % (provided the threshold corresponds to a low false-alarm level), and we find generally good agreement (of less than $\sim 10\%$ average error) compared to published upper limits in the literature. Several factors leading to the observed deviations in various cases are discussed in detail.

We also provided a comprehensive overview of published CW upper limit results, converting the quoted h_0 upper limits into sensitivity depths. This introduces some systematic uncertainties, as we often do not have access to the original PSD estimate used for the upper limits. We therefore advocate for future searches to directly provide their upper-limit results also in terms of the sensitivity depth of Eq. (6.28), in order to allow easier direct comparison between searches and to sensitivity estimates.

6.8. Appendix A: Details on Referenced CW Searches

6.8.1. General Remarks

In this appendix we will refer to the different detectors as G for GEO600 [47], V for VIRGO [45, 46], H1 and H2 for the two LIGO detectors in Hanford (4km, 2km) and L1 for LIGO Livingston [43, 44].

We will use the common abbreviations CW for continuous gravitational waves, SFT for Short Fourier Transform, PSD for power spectral density and UL for upper limits.

The quoted sensitivity depths in Tables 6.1 - 6.5 can correspond to different confidence levels, as some searches use 90%- and others 95%-confidence upper limits. The applicable confidence level is denoted by using regular versus italic font in the tables, respectively.

For searches over many frequencies, multiple targets or for upper limits reported separately for different detectors, we use a consistent averaging procedure using the median and median absolute deviation of Eq. (6.55) in order to estimate the mean and standard deviation in an outlier-robust way.

PowerFlux and loosely-coherent searches typically give separate upper limits for circular (best) polarisation and for the worst linear polarization, but not the more common type of population-averaged upper limits. There has been some work estimating conversion factors for these upper limits into polarization-averaged sensitivity, writing $\mathcal{D}^{\text{PF}} \sim w_{\text{worst}} \mathcal{D}_{\text{worst}}^{\text{PF}}$ and $\mathcal{D}^{\text{PF}} \sim w_{\text{best}} \mathcal{D}_{\text{best}}^{\text{PF}}$. For example [29] obtains the conversion factors in the ranges $w_{\text{worst}} \sim 1.1 - 1.3$ and $w_{\text{best}} \sim 0.39 - 0.46$. More recent work estimating these conversion factors on O1 data (cf. Fig.[188]) for 90%-confidence upper limits yields [232] $w_{\text{worst}} = 1.51 \pm 0.13$ and $w_{\text{best}} = 0.52 \pm 0.02$. However, these conversion factors were obtained by treating the set of upper limits as a whole, they should not be used to derive a proxy of population average upper limits in individual frequency bands. Furthermore, PowerFlux strict upper limits are derived by taking the highest upper limits over regions of parameter space. This procedure has the advantage of the upper limits retaining validity over any subset of parameter space, such as a particular frequency or a particular sky location. However, the maximization procedure makes it difficult to convert the data into population average upper limits which are more robust to small spikes in the data. Given that there is currently some uncertainty on the detailed values of the conversion factors

to use for different PowerFlux searches, here we report the best/worst upper limits converted into sensitivity depths separately in Tables 6.1 and 6.2.

Generally, for converting h_0 upper limits into depths according to Eq. (6.28), we need to use an estimate for the corresponding noise PSD \mathcal{S} , for which we either use a corresponding PSD over the data used in the search, where available, or a 'generic' PSD estimate from LIGO for the given science run [233, 234] otherwise. This adds another level of uncertainty in the conversions, which could easily be in the range 10% – 20% due to different calibrations and different types of averaging over time.

6.8.2. All-sky Searches, see Table 6.1

S2-AllSky-Hough [209]

The first all-sky search for CWs from isolated neutron stars, using a semi-coherent Hough transform method applied on Short Fourier Transforms (SFTs) of the data of length $T_{\text{seg}} = 30$ min. The search used data from the second LIGO Science Run (S2), and the number of SFTs used in the search was 687 from L1, 1761 from H1 and 1384 from H2.

The UL sensitivity depth for this search is calculated as the mean over the three depths for H1, L1 and H2, where each depth is computed from the respective quoted best upper-limit value $h_0^{95\%}$ and the corresponding PSD \mathcal{S} in TABLE III of [209].

S2-AllSky- \mathcal{F} [210]

A matched-filtering search based on the coherent (single-detector) \mathcal{F} -statistics, using 20 SFTs from H1 and 20 SFTs from L1 (SFT length $T_{\text{SFT}} = 30$ min). The per-detector \mathcal{F} -statistic values were combined via a coincidence scheme, determining the most significant candidate in each ~ 1 Hz band, which was then used for measuring the upper limits.

The sensitivity depth for this search is calculated from the given (combined multi-detector) upper limits $h_0^{95\%}(f)$ over the search frequency range, combined with the harmonic mean over generic H1- and L1- PSDs for the LIGO S2 data.

The estimate was calculated with the mean loudest templates of the search given in the paper as $\mathcal{F}_{\text{th}} = (39.5, 32.2)$ for the L1 and H1 detector, respectively, and we used an average mismatch of 0.5% in the H1 search and 1% in the L1 search, estimated from Figs. 27, 28 in [210].

S4-AllSky-{StackSlide,Hough,PowerFlux}[195]

Three semi-coherent all-sky searches using different search methods, all based on incoherently combining SFTs of length $T_{\text{seg}} = 30$ min. The StackSlide and the Hough search used 1004 SFTs from H1 and 899 from L1 and the Hough search additionally included 1063 SFTs from H2. The PowerFlux search used 1925 and 1628 SFTs from H1 and L1, respectively.

The sensitivity depths are calculated from the quoted upper limits $h_0^{95\%}(f)$ from each of the three searches over the search frequency range, combined with the PSDs for two (H1 and L1) detectors (as a common reference) from the S4 science run. Note that the Hough depth corresponds to the quoted multi-detector UL, while the other searches reported only per-detector ULs.

S4-AllSky- \mathcal{F} +Coinc[211]

A search which used the distributed computing project Einstein@Home [7] to analyze 300 h of H1 data and 210 h of L1 data from the S4 run. The data was split into 30 h long segments coherently analyzed with the multi-detector \mathcal{F} -statistic followed by a coincidence-step. The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%}$ is calculated by converting the quoted sensitivity factors $R_{90\%} = \{31.8, 33.2\}$ (for frequencies below and above 300 Hz, respectively) into sensitivity depths. However, given

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

these were computed with respect to an (arithmetic) averaged PSD estimate (given in Fig.1 in the paper), we first converted these factors back into equivalent h_0 values using the mean-PSD, and then computed the Depth with respect to the harmonic-mean (over detectors) generic noise PSD for S4.

earlyS5-AllSky-PowerFlux [212]

An all-sky search with PowerFlux over the first eight months of S5 data. The search in total used roughly 4077 h of H1 data and 3070 h L1 data, divided into SFT segments of $T_{\text{seg}} = 30$ min.

The sensitivity depth is calculated from the quoted per-detector upper limits $h_0^{95\%}(f)$ over the search frequency range and the corresponding S5 noise PSDs.

earlyS5-AllSky- \mathcal{F} +Coinc [213]

An all-sky search run on Einstein@Home [7], using 660 h of data from H1 and 180 h of L1 data, taken from the first 66 days of the LIGO S5 science run. The data was divided into 28 segments of $T_{\text{seg}} = 30$ h duration, and each segment was searched using the fully-coherent multi-detector \mathcal{F} -statistic. These per-segment \mathcal{F} -statistics were combined across segments using a coincidence scheme.

The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%}$ is calculated as the median over the converted sensitivity depths converted from the quoted sensitivity factors $R_{90\%} = \{29.4, 30.3\}$ in the paper for the frequencies below and above 400 Hz, respectively.

S5-AllSky-PowerFlux [214]

An all-sky search using PowerFlux analyzing the whole of LIGO S5 data, broken into more than 80 000 50%-overlapping 30-minute SFTs from both H1 and L1.

The sensitivity depth is calculated from the quoted upper limits $h_0^{95\%}$ and the S5 noise PSD.

S5-AllSky-Hough- \mathcal{F} [196]

An all-sky search using the Hough- \mathcal{F} variant of the semi-coherent Hough method described in Sec. 6.2.3, which was run on Einstein@Home. The analyzed data consisted of 5550 and 5010 SFTs from the LIGO H1 and L1 interferometers, respectively, taken from the second year of the S5 science run. The data was divided into 121 segments of length $T_{\text{seg}} = 25$ h, and the coherent per-segment \mathcal{F} -statistic was combined via the Hough method to compute the Hough number count of Eq. (6.25).

The sensitivity depth of the search is calculated from the quoted $h_0^{90\%}$ upper limits and the corresponding S5 noise PSD.

The estimated sensitivity depth uses the generalization of the estimator described in Sec. 6.3.5 with a number-count threshold of $n_{c,\text{th}} = 70$, a per segment threshold of $\tilde{\mathcal{F}}_{\text{th}} = 2.6$ and a mismatch histogram obtained from an injection-recovery simulation (with an average mismatch of $\tilde{\mu} = 0.61$).

S5-AllSky-Hough [215]

An SFT-based Hough all-sky search on S5 data. The search was split into the first and the second year of S5, which were searched separately. The first year used 11 402 SFTs from H1, 12 195 SFTs from H2 and 8 698 SFTs from L1, of length $T_{\text{SFT}} = 30$ min. The analysis of the second year used 12 590 H1-SFTs, 12 178 H2-SFTs and 10 633 L1-SFTs.

The sensitivity depth is calculated from the quoted $h_0^{90\%}$ upper limits of the second year search found in the paper and from the S5 noise PSD.

S5-AllSky-StackSlide- \mathcal{F} [216]

A high frequency all-sky search to complement previous lower-frequency all-sky searches on S5 data. The search used the so-called GCT method [138] implementing the StackSlide- \mathcal{F} statistic and was run on the distributed Einstein@Home platform. The search used a total of 17797 SFTs spanning the whole two years of S5 data from H1 and L1, divided into 205 segments of length $T_{\text{seg}} = 30$ h.

The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%}$ is determined by extrapolating the depth values given in the paper for critical ratios of 0 and 3.5 to the median critical ratio over all frequency bands of -0.15 according to figure 6 of [216].

For the estimate we determined the median threshold over all frequency bands from figure 4 of [216] to $2\bar{\mathcal{F}}_{\text{th}} = 5.72$. Two mismatch histograms at 1255 Hz and 1495 Hz generated with injection-recovery studies were used. The average mismatch for both was $\mu \approx 0.82$. The quoted value is the mean of the two estimates with different mismatch histograms.

VSR1-AllSky- \mathcal{F}_{TD} +Coinc [217]:

An all-sky search using data from the first Virgo science run, VSR1. The search method uses a time-domain implementation of the coherent \mathcal{F} -statistic, computed over 2-day coherent segments, which are combined using coincidences. In total the search used 134 days of data.

The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%}$ is calculated as median of the given sensitivity factors of 15.6 and 22.4.

{VSR2, 4}-AllSky-FreqHough+FUP [218]

This all-sky search was performed using data from initial Virgos second (VSR2) and fourth (VSR4) science run. It used the FrequencyHough transform as incoherent step with 149 days of data of VSR2 and 476 days of data of VSR4 using segments of length 8192 seconds. The initial candidates were followed-up using 10 times longer segments.

The measured sensitivity depth was calculated from upper limits $h_0^{90\%}$ extracted from figure 12 of [218] and the harmonic mean of the PSD estimates of VSR2 and VSR4 in 0.1 Hz frequency bands.

S6-AllSky-StackSlide- \mathcal{F} [207]

This search used 12080 SFTs from L1 and H1 data to perform a StackSlide- \mathcal{F} search based on the GCT implementation, and was run on Einstein@Home. The search used 90 coherent segments of length $T_{\text{seg}} = 60$ h.

The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%}$ is determined by extrapolating the depth from the given critical ratios 0 and 6 to the median critical ratio of -0.07 according to figure 5 of [207].

The estimated depth is given for a threshold of $2\bar{\mathcal{F}}_{\text{th}} = 6.694$ which is the median of the thresholds given for the frequency bands in figure 4 of [207]. For the estimate two mismatch histogram created with injection-recovery studies for 55 Hz and 505 Hz was used. The average mismatch of the grid in the parameter space was at both frequencies found to be $\mu = 0.72$. The quoted value is the mean of the two estimates with different mismatch histograms.

S6-AllSky-StackSlide- \mathcal{F} +FUP [201]

A multi-stage follow-up on candidates from the S6-AllSky-StackSlide- \mathcal{F} search described in the previous paragraph, zooming in on candidates using increasingly finer grid resolution and longer segments. Every candidate from the initial stage with $2\bar{\mathcal{F}} \geq 6.109$ was used as the center of a new search box for the first-stage follow-up, continuing for a total of four semi-coherent follow-up stages. The sensitivity of the search is dominated by the initial-stage threshold,

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

because the later stages are designed to have a very low probability of dismissing a real signal. The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%} = 46.9 \text{ Hz}^{-1/2}$ of this search is directly taken from the quoted value in the paper.

The estimated multi-stage sensitivity of Sec. 6.3.4 using the thresholds given in the paper, namely $\{2\overline{\mathcal{F}}_{\text{th}}^{(i)}\} = (6.109, 6.109, 7.38, 8.82, 15)$ and a mismatch histogram generated by recovery injection studies for the main search and mismatch histograms provided by the original authors for every stage with average mismatches $\{\mu^{(i)}\} = (0.72, 0.55, 0.54, 0.29, 0.14)$, yields a value of $\mathcal{D}^{90\%} = 38.3 \text{ Hz}^{-1/2}$, which differs significantly from the quoted measured sensitivity depth. As discussed in Sec. 6.5, we trace this discrepancy to the low threshold used, which significantly affects the loudest-candidate mismatch approximation used in the theoretical estimate.

S6-AllSky-PowerFlux [219]

The data used by this search span a time of 232.5 d with duty factor of the detectors of 53% for H1 and 51% for L1.

The measured sensitivity depth is calculated from the quoted upper limits $h_0^{95\%}$ in the paper and the S6 noise PSD.

O1-AllSky-StackSlide- \mathcal{F} [188]

A low-frequency all-sky search for gravitational waves from isolated neutron stars using the distributed computing project Einstein@Home on data from Advanced LIGO's first observing run (O1). This search used the GCT implementation of the semi-coherent StackSlide- \mathcal{F} method with $N_{\text{seg}} = 12$ segments of length $T_{\text{seg}} = 210 \text{ h}$ in the initial search stage. The analyzed data consisted of 4744 SFTs from the H1 and the L1 detector. The search also included a hierarchical follow-up similar to the S6Bucket follow-up search[201].

The measured sensitivity depth $\mathcal{D}_{\text{meas}}^{90\%} = 48.7 \text{ Hz}^{-1/2}$ of this search is directly taken from the quoted value in the paper.

The sensitivity estimate used a threshold $2\overline{\mathcal{F}}_{\text{th}} = 14.5$ which we inferred from figure 4 in [188] and we obtained the mismatch histograms of the template grid at different frequencies using an injection-recovery study, which yielded an average mismatch of $\mu = 0.35$ and $\mu = 0.37$ at 20 Hz and 100 Hz respectively. The quoted depth is the average of the two different estimates resulting for each mismatch histogram. Note that the contrary to the measured sensitivity, the estimate only uses the first-stage parameters in this case, as we currently cannot model the line-robust statistic used in the follow-up stages. However, as mentioned in Sec. 6.3.4, the overall detection probability is dominated by the first stage, while subsequent stages mostly serve to reduce the false-alarm level.

O1-AllSky-{PowerFlux, Hough, $\mathcal{F}_{\text{TD+Coinc}}$ } [127, 187]

Two papers detailing the results of all-sky searches on O1 data using four different search methods.

The first paper [187] searched the lower frequency range [20, 475] Hz, using four methods: PowerFlux, FrequencyHough, SkyHough and a time-domain \mathcal{F} -statistic search with segment-coincidences (denoted as $\mathcal{F}_{\text{TD+Coinc}}$). The PowerFlux, FrequencyHough and SkyHough search used SFT lengths in the range 1800 – 7200s as coherent segments while the Time-Domain \mathcal{F} -statistic used a coherence time of $T_{\text{seg}} = 6 \text{ d}$. The total amount of analyzed data was about 77 d of H1 data and 66 d of L1 data.

In the second paper [127] three of these searches were extended up to 2000 Hz, namely PowerFlux, SkyHough and a time-domain \mathcal{F} -statistic search with segment-coincidences (denoted as $\mathcal{F}_{\text{TD+Coinc}}$), using the same data.

The sensitivity depths for the four searches are calculated from the quoted $h_0^{95\%}$ amplitude upper limits and the noise PSD for the O1 science run.

Note that for the SkyHough method a sensitivity depth of $24.2 \text{ Hz}^{-1/2}$ is quoted in the paper. However, this value is based on a slightly different convention for the multi-detector noise PSD \mathcal{S} (maximum over detectors instead of the harmonic mean) than used here. For consistency with the other searches in Table 6.1 we therefore compute the sensitivity depth by converting from the quoted $h_0^{95\%}$ upper limits instead.

A comparison of *PowerFlux* 90%-confidence upper limits for an *isotropic polarization* population were provided for the O1 Einstein@Home paper [188], with a frequency spacing of 0.0625 Hz, which are converted into sensitivity depth using the O1 noise PSD.

6.8.3. Directed Searches, see Tables 6.2, 6.3

earlyS5-Crab- \mathcal{F} [221]

This search aimed at the Crab pulsar and used the first nine month of initial LIGO’s fifth science run (S5). It consisted of both a targeted (described in Sec. 6.8.5) and a directed \mathcal{F} -statistic search described here. The directed search used 182, 206 and 141 days of data from the H1, H2 and L1 LIGO detectors, respectively. The measured depth value is calculated from the given upper limits $h_0^{95\%}$ and the PSD estimate of the S5 data at the search frequency.

The estimated depth uses the StackSlide estimator for a coherent search with $N_{\text{seg}} = 1$ segment, a threshold of $\mathcal{F}_{\text{th}} = 37$ and a maximal template bank mismatch of 5% (given in the paper), from which we estimate the average mismatch as $\tilde{\mu} \sim \frac{1}{3} 5\%$ (assuming a square lattice).

S5-CasA- \mathcal{F} [200, 235]

The first search for continuous gravitational waves from the Cassiopeia A supernova remnant using data from initial LIGO’s fifth science run (S5). The search coherently analyzed data in an interval of 12 days (934 SFTs of length 30 min) using the \mathcal{F} -statistic.

The measured sensitivity depth is obtained from the quoted upper limits $h_0^{95\%}$ in the paper and the S5 noise PSD.

The estimate is calculated using the StackSlide estimator for a coherent search ($N_{\text{seg}} = 1$ segment), with the mismatch histogram for an A_n^* lattice with maximal mismatch of $\mu = 0.2$ (obtained from `LATTICEMISMATCHHIST()` in [205]), and the average threshold of $2\mathcal{F}_{\text{th}} = 55.8$ (averaged over the respective loudest $2\mathcal{F}$ -candidates found in each of the upper-limit bands).

S5-GalacticCenter-StackSlide- \mathcal{F} [199, 220]

The first search for continuous gravitational waves directed at the galactic center. The search used LIGO S5 data and the GCT implementation of the StackSlide- \mathcal{F} semi-coherent search algorithm with 630 segments, each spanning 11.5 h, for total data set of 21 463 SFTs of length 30 min.

The segments of the search were selected from the whole S5 science run in such a way as to maximize the SNR for fixed-strength GW signals at the sky position of the galactic center. Therefore the selected segments fall at times where the antenna patterns of the LIGO detectors are better than average for this particular skyposition. As discussed in Sec. 6.6.3, the sensitivity-estimation method presented in this work assumes the antenna patterns are averaged over multiple days, which causes a unusually large deviation between the estimate and the measured sensitivity depth from the $h_0^{90\%}$ upper limits.

The estimate is calculated using the mismatch histogram (with mean $\mu = 0.13$) obtained from an injection-recovery study on the template bank of this search, and a detection threshold of $2\overline{\mathcal{F}}_{\text{th}} = 4.77$.

VSR4-{Vela,Crab}-5-vector [222]

This coherent narrow-band search on the data from initial Virgo's forth science run (VSR4) was directed at the Vela and the Crab pulsars. This search used the 5-vector method, and covers a range of ± 0.02 Hz the twice the known frequencies of Vela and Crab. The total amount of data used is 76 d.

The measured sensitivity depth for this search was obtained from the published $h_0^{95\%}$ upper limits and the noise PSD estimate for VSR4.

S6-NineYoung- \mathcal{F} [184]

This search was directed at nine different targets, listed in Table 6.3, each corresponding to a (confirmed or suspected) compact object in a young supernova remnant. The search uses a fully-coherent \mathcal{F} -statistic. The amount of data used for every target varies between $7.3 \cdot 10^5$ s and $3.1 \cdot 10^6$ s (cf. Table 6.3).

The measured depth is calculated for each of the targets from the quoted upper limits $h_0^{95\%}$ and the corresponding PSD for the actual data used in the search.

The estimate for each target is calculated using the StackSlide estimator for a coherent search ($N_{\text{seg}} = 1$ segment), with the mismatch histogram for an A_n^* lattice with maximal mismatch of $\mu = 0.2$ (obtained from `LATTICEMISMATCHHIST()` in [205]), and the average $2\mathcal{F}_{\text{th}}$ threshold found for each target (averaged over the respective loudest $2\mathcal{F}$ -candidates found in each of the upper-limit bands) are given in Table 6.3.

The 'NineYoung' entry in Table 6.2 presents the median depth over all targets for the measured and estimated depths, respectively.

S6-CasA-StackSlide- \mathcal{F} [185]

A search directed at Cassiopeia A, which was run on the distributed computing project Einstein@Home using data from the LIGO S6 science run. The search was based on the GCT implementation of the semi-coherent StackSlide- \mathcal{F} statistic, with $N_{\text{seg}} = 44$ segments of length $T_{\text{seg}} = 140$ h, and a total amount of data of 13 143 SFTs of length 30 min from the two LIGO detectors in Hanford (H1) and Livingston (L1). The measured sensitivity depth given in Table 6.2 is computed from the $h_0^{90\%}$ upper limits quoted the paper [185] combined with the corresponding PSD estimates. However, as discussed in 6.6.3, this measurement suffered from a bug in the upper-limit script and as a result is somewhat too conservative (i.e. too high).

The estimated sensitivity is calculated assuming an average threshold of $\bar{\mathcal{F}}_{\text{th}} = 8.25$ (estimated from Fig. 4 in [185]) using the mean over estimates with different mismatch histograms generated by injection-recovery studies at different frequencies (spanning 50 – 1000 Hz, average mismatch $\sim 9\%$).

S6-OrionSpur-LooselyCoherent [223]

This was a search employing the so-called loosely-coherent method, aimed at the Orion spur towards both the inner and outer regions of our Galaxy. The explored sky regions are disks with 6.87° diameter around $20^{\text{h}}10^{\text{m}}54.71^{\text{s}} + 33^\circ 33' 25.29''$ and 7.45° diameter around $8^{\text{h}}35^{\text{m}}20.61^{\text{s}} - 46^\circ 49' 25.151''$. The data used in this search spanned 20 085 802 s with duty factors of 53% and 51% for LIGO Hanford and Livingston respectively. Due to weighting of the data the effective amount of data used was only $\sim 12.5\%$ of the available S6 data. For the analysis data segments of length 30 min were searched coherently.

The measured sensitivity depth was calculated from the quoted upper limits $h_0^{95\%}$ and a PSD estimate for the LIGO S6 data.

S6-NGC6544- \mathcal{F} [224]

This was the first search directed at the nearby globular cluster NGC 6544. The search coherently analyzed data from the two LIGO detectors S6 science run with the \mathcal{F} -statistic, using a single coherent segment with $T_{\text{seg}} = 9.2$ d. The search analyzed two different data stretches separately. The first one contained 374 SFTs while the second contained 642 SFTs, with SFT duration of 30 min.

The measured depth was determined from the upper limits $h_0^{95\%}$ given in figure 2 of [224] and a PSD estimate for the LIGO S6 run.

The estimate used the StackSlide estimator with one segment, a threshold of $2\mathcal{F}_{\text{th}} = 55$ (quoted in the paper) and an average mismatch of $0.2/3$ (assuming a roughly square lattice).

01-Narrow-band-5-vector[183]

A narrow-band search aiming at 11 known pulsars using the fully-coherent 5-vector method on data from Advanced LIGO's first observing run (O1). The search used a total of 121 days of data from the Hanford (H1) and Livingston (L1) detectors.

The sensitivity depth in the table is calculated from the median over the single-target depths, which are converted from the upper-limits $h_0^{95\%}$ quoted in the paper and the corresponding noise PSD of the data used.

01-{SN1987, GalacticCenter}-Radiometer[225]

Described in Sec.6.8.4.

6.8.4. Searches for Neutron Stars in Binary Systems, see Table 6.4**S2-ScoX1- \mathcal{F} [210]**

This first search designed specifically aimed at the NS in the LMXB system Scorpius X-1, using a coherent single-detector \mathcal{F} -statistic and a coincidence check on a 6 h long stretch of S2 dat.

The measured sensitivity depth was calculated from the quoted upper limits $h_0^{95\%}$ in the paper (for the zero-eccentricity case $e = 0$) and the PSD estimate of the corresponding S2 data.

S5-ScoX1-Sideband[226]

A search aimed at Scorpius X-1 by incoherently combining sidebands of a coherent \mathcal{F} -statistic search that only demodulates the signal for the sky-position but not its binary-orbital Doppler modulation. This method used a stretch of 10 days of data selected from the S5 science run for maximal sensitivity. Two searches were performed, one with no prior assumptions about the orientation of Sco-X1, and one using more restrictive angle-priors based on electromagnetic observations.

Bayesian upper limits $h_0^{95\%}$ were computed over the search frequency range, which we convert into sensitivity depths (for the unknown-polarization case, see Fig.5(a) in [226]) using the noise PSD for the data given in the paper. In each 1Hz-band, 2×10^6 upper limit values were quoted, of which we use the maximum value in each 1Hz-band in order to be consistent with the usual “loudest-candidate” approach of setting upper limits in a given frequency band.

{S6, VSR2, 3}-{AllSky, ScoX1}-TwoSpect[189]

A TwoSpect search for unknown binary signals from any sky-position, and a directed TwoSpect search for Scorpius X-1 specifically. This search used data from LIGO S6 science run, as well as from Virgo VSR2 and VSR3 runs, spanning 40 551 300 s from each detector.

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

The quoted upper limits $h_0^{95\%}$ for the all-sky search and the Scorpius X-1 search were converted into Depths using a combined (generic) PSD for the S6, VSR2 and VSR3 science runs.

S6-{ScoX1, J1751}-TwoSpect[227]

A search for CW from the low-mass X-ray binaries Scorpius X-1 and XTE J1751-305 using the TwoSpect algorithm. It used about $4 \cdot 10^7$ s from each of the two detector in the S6 science run. It used two different length of the SFTs 840 s and 360 s which also where the length of the coherently analysed segments.

The given sensitivity depth $\mathcal{D}_0^{95\%}$ is obtained from the quoted $h_0^{95\%}$ upper limits combined with the corresponding noise PSD for S6 data.

O1-ScoX1-Viterbi[186]

A search aimed at Scorpius X-1 using the Viterbi search method performed on 130 days of data from Advanced LIGO's first observational run (O1), segmented into coherent segments of length $T_{\text{seg}} = 10$ days.

The measured sensitivity depth is converted from the quoted upper limits $h_0^{95\%}$ (for unknown polarization) and the noise PSD of the corresponding O1 data.

Note that contrary to many other search methods, this search setup appears to result in a frequency-dependent sensitivity depth, namely $\mathcal{D}(f) \propto f^{-1/4}$ (see Eq.(9) in [186]). For consistency with other searches, we quote the median and (MAD) standard-deviation over frequencies in Table 6.4, and note that the total range of sensitivity depths of this search is found as $\mathcal{D}(f) \sim 11 (f/f_0)^{-1/4} \text{ Hz}^{-1/2} \in [4.6, 11.2] \text{ Hz}^{-1/2}$ with $f_0 = 60.5 \text{ Hz}$.

O1-ScoX1-CrossCorr[135]

This search aimed at Scorpius X-1 using the CrossCorr search algorithm using data from Advanced LIGO's first observational run (O1). The data was split into coherently analysed segments (SFTs) with a (frequency-dependent) length between 240 s and 1400 s.

The measured sensitivity depth is obtained from the quoted (isotropic-prior) upper limits $h_0^{95\%}$ and the noise PSD of the O1 data. Note, however, that the search ULs are given per 0.05 Hz bands, which is unusually small compared to most other upper-limit bands (typically 0.25 – 1 Hz), and therefore they display more variability. In order to make these ULs more comparable to other searches, we use the 95th-percentile highest upper limits per 1Hz-bands (as recommended in Fig. 5 of [135]). This 'binning' procedure only has a small effect on the resulting sensitivity depth, which is reduced from $25.3 \text{ Hz}^{-1/2}$ to $24.0 \text{ Hz}^{-1/2}$.

Note that this search has a frequency-dependent sensitivity depth, which starts at around $\mathcal{D}(25 \text{ Hz}) \sim 45 \text{ Hz}^{-1/2}$ for low frequencies, asymptoting down to $\mathcal{D} \sim 23 \text{ Hz}^{-1/2}$ above $f \gtrsim 800 \text{ Hz}$. However, in order to be consistent with other searches, we quote the median and (MAD) standard deviation over all frequencies in Table. 6.4.

O1-{ScoX1 and others}-Radiometer[225]

The 'Radiometer' search method, which was developed mainly for stochastic background searches, can also be used for directed CW searches at particular sky-positions. This method does not use a particular signal model, which allows it to be sensitive to a wide range of possible signal families, at the cost of somewhat lower sensitivity to 'regular' CW signals. This search aimed at the sky-positions of Sco-X1, as well as at the supernova remnant 1987A and the galactic center.

The search reported $h_0^{90\%}$ (and $h_0^{95\%}$ for Sco-X1, reported in [135]) upper limits in narrow frequency bands of $1/32 \text{ Hz} = 0.03125 \text{ Hz}$ bands, which is unusually small compared to most other upper-limit bands (typically 0.25 – 1 Hz), and therefore they display more variability.

In order to make these ULs more comparable to other searches, we use the 95th-percentile highest upper limits per 1Hz-bands (as recommended in Fig. 5 of [135]), and following the same procedure as used for the CrossCorr results (discussed in Sec. 6.8.4).

6.8.5. Targeted Searches, see Table 6.5

S1-J1939+21- $\{\mathcal{F}, \text{BayesPE}\}$ [193]

This first CW search on data from GEO 600 and LIGO's first science run (S1). It used (16.7, 5.73, 8.73, 8.9) days of data from four detectors, GEO 600 (G1), LIGO Livingston (L1), LIGO Hanford-4 km (H1), and LIGO Hanford-2 km (H2), respectively. Two types of searches were performed, a coherent \mathcal{F} -statistic search as well as direct Bayesian parameter estimation (BayesPE).

Table 6.5 gives the mean and standard deviation for the sensitivity depths over the four detectors. The measured sensitivity depth for the \mathcal{F} -search was determined from the quoted upper limits $h_0^{95\%}$ in table IV[193] for the most pessimistic ι ($\cos \iota = 0$) and ψ , and from the quoted numbers in the conclusion for the (standard) population-averaged orientation. The noise PSD values are taken from table III in [193]. The corresponding estimate is calculated with the StackSlide estimator for $N_{\text{seg}} = 1$ and quoted threshold values $2\mathcal{F}_{\text{th}} = (1.5, 3.6, 6.0, 3.4)$ for the four detectors from Table III in the paper. For the 'worst-case' estimate we use the prior $\cos \iota = 0$ and minimise the sensitivity depth over $\psi \in [-\pi/4, \pi/4]$ in order to reflect the 'conservative' ULs quoted in the paper. Note, however, that contrary to the typically small false-alarm level (p-value) of the UL thresholds used (typically 1%), the loudest candidates used as thresholds had relatively high p-values of 83%, 46%, 20% and 49%, respectively, as seen in table III of [193].

S2-Known pulsars-BayesPE[228]

A coherent targeted search for 28 known isolated radio pulsars was performed using the Bayesian parameter-estimation pipeline (BayesPE) on data from the second LIGO Science Run (S2), using 910 h of data from H1, 691 h from H2 and 342 h of L1 data from the S2 data set.

The measured sensitivity depth is calculated from the quoted Bayesian upper limits $h_0^{95\%}$ and corresponding noise PSD estimates for the S2 science run.

The sensitivity estimate is performed using the Bayesian sensitivity estimator, for simplicity assuming the sources are distributed isotropically over the sky.

$\{\text{S3}, 4\}$ -Known pulsars-BayesPE[229]

This search targeted 78 known radio pulsars by analysing (45.5, 42.1, 13.4) days of data from the three detectors (H1, H2, L1) from the third science run (S3) of LIGO and GEO 600, and (19.4, 22.5, 17.1) days of data from the three detectors from the S4 science run. The analysis used the Bayesian parameter-estimation pipeline (BayesPE).

The measured sensitivity depth was determined from the quoted Bayesian upper limits $h_0^{95\%}$ combined with the noise PSD of the S3 and S4 science runs combined (using harmonic mean).

The sensitivity estimate is calculated using the Bayesian sensitivity estimate, for simplicity assuming the sources to be isotropically distributed on the sky.

earlyS5-Crab-BayesPE[221]

This search on 9 months of data from the early LIGO S5 science run targeted only the Crab pulsar at twice its rotation rate, using the Bayesian parameter-estimation pipeline. A corresponding narrow-band search using the \mathcal{F} -statistic is described in Sec. 6.8.3. The targeted search used 201, 222 and 158 days of data of the H1, H2 and L1 LIGO detectors.

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

The measured depth is determined from the quoted (i.e. the corrected value in the Erratum) upper limit $h_0^{95\%}$ assuming an isotropic polarization prior, and the corresponding noise PSD of the detectors for the early S5 science run data.

S5-Known pulsars-BayesPE[230]

A search targeting 116 known pulsars using 525 days of H1 data, 532 days of H2 data and 437 days of L1 data from LIGO's fifth science run (S5). The search employed the Bayesian parameter-estimation pipeline.

The measured sensitivity depth is calculated from the quoted Bayesian upper limits $h_0^{95\%}$ and the noise PSD of the S5 data.

The estimate is calculated with the Bayesian sensitivity estimator under the assumption that the targets are distributed isotropically over the sky.

VSR2-Vela-{BayesPE, \mathcal{F} , 5-vector}[231]

A targeted search for the Vela pulsar using Virgo's second science-run (VSR2) data, using three different methods: Bayesian parameter estimation, the \mathcal{F} -statistic (and \mathcal{G} -statistic) and the 5-vector method. The data set consisted of 149 days of Virgo data.

Two types of searches and upper limits were computed, namely (i) using uninformative (isotropic) priors on the pulsar orientation, and (ii) using angle priors on $\cos\iota$ and ψ from electromagnetic observations.

In Table 6.5 we only give the measured depth corresponding to the isotropic prior, averaged over the three methods, which obtained very similar results. This was computed from the quoted upper limits $h_0^{95\%}$ and the noise PSD for the Vela VSR2 run. The measured sensitivity depth obtained when using the angle priors is found as $DVelaAnglePriors \text{ Hz}^{-1/2}$.

The estimated sensitivity depth is calculated using the Bayesian sensitivity estimator.

{S6, VSR2, 4}-Known pulsars-{BayesPE, \mathcal{F} , 5-vector}[203]

This search targeted 195 known pulsars, using 149 days of VSR2 and 76 days of VSR4 data for pulsars with a CW frequency lower than $f < 40$ Hz and an additional 238 days of S6 data from H1 and 225 days from L1 for faster spinning pulsars with $f > 40$ Hz. The analysis was done using three different methods: Bayesian parameter estimation, the \mathcal{F} -statistic (or \mathcal{G} -statistic for restricted angle priors) and the 5-vector method.

The given measured sensitivity depth in Table 6.5 is the median and MAD standard deviation over the sensitivity depths for the different targets (averaged over high- and low-frequency targets). The sensitivity depths are obtained from the quoted upper limits $h_0^{95\%}$ and the corresponding noise PSD estimate of the data used (which is either S6 and VSR2 and VSR4 for high-frequency targets $f > 40$ Hz, or only VSR2 and VSR4 for low-frequency targets).

The estimated sensitivity is obtained from the Bayesian sensitivity estimator assuming an isotropic prior over the sky, averaged over high- and low-frequency depths results.

O1-Known pulsars-{BayesPE, \mathcal{F} , 5-vector}[182]

In this search 200 known pulsars were targeted using three different methods: Bayesian parameter estimation, the \mathcal{F} -statistic (or \mathcal{G} -statistic for restricted angle priors) and the 5-vector method. The searches used 78 and 66 days of H1 and O1 data from the first observational run of advanced LIGO (O1), respectively.

The measured sensitivity depth is obtained from the quoted Bayesian upper limits $h_0^{95\%}$ over all targets and the corresponding noise PSD for the LIGO detectors during O1.

The estimated sensitivity depth is determined from the Bayesian estimator as an all-sky estimate assuming the targets are isotropically uniformly distributed over the sky.

6.9. Appendix B: CW Signal Model and \mathcal{F} -statistic

A plane gravitational wave arriving from a direction \hat{n} (unit vector) can be written [236] in TT gauge (in the notation of [105]) as a purely spatial strain tensor \overleftrightarrow{h} with two polarizations $+$, \times , namely

$$\overleftrightarrow{h}(\tau) = h_+(\tau) \overleftrightarrow{e}_+ + h_\times(\tau) \overleftrightarrow{e}_\times, \quad (6.56)$$

where τ is the emission time of the signal in the source frame, and \overleftrightarrow{e}_+ and $\overleftrightarrow{e}_\times$ are the two polarization basis tensors, which can be constructed from a right-handed orthonormal basis $\{\hat{\ell}, \hat{m}, -\hat{n}\}$ as $\overleftrightarrow{e}_+ = \hat{\ell} \otimes \hat{\ell} - \hat{m} \otimes \hat{m}$ and $\overleftrightarrow{e}_\times = \hat{\ell} \otimes \hat{m} + \hat{m} \otimes \hat{\ell}$.

The measured scalar CW signal $h^X(t)$ at time t by detector X is the response of the detector to the GW tensor $\overleftrightarrow{h}(\tau^X(t))$, where $\tau^X(t)$ denotes the emission time of a wavefront that reaches detector X at time t . This timing relationship depends on the sky-position \hat{n} of the source as well as any binary-orbital parameters in case of a CW from a neutron star in a binary system, as it describes the time-dependent light-travel time from the source to the detector. In the long-wavelength limit we assume the GW wavelength to be much larger than the detector armlength, which is a good approximation for current ground-based detectors up to kHz frequencies. This allows us to write the detector response as a tensor contraction (in both tensor indices):

$$h^X(t) = \overleftrightarrow{d}^X(t) : \overleftrightarrow{h}(\tau^X(t)), \quad (6.57)$$

where $\overleftrightarrow{d}^X = \hat{u} \otimes \hat{u} - \hat{v} \otimes \hat{v}$ for interferometer arms along unit vectors \hat{u} and \hat{v} .

It is helpful to define a *source-independent* orthonormal polarization basis $\{\hat{i}, \hat{j}, -\hat{n}\}$ instead, where for any sky position \hat{n} , the unit vector \hat{i} is chosen to lie in Earth's equatorial plane (pointing West) and \hat{j} is pointing in the northern hemisphere. This defines the (sky-position dependent) alternative polarization basis as $\overleftrightarrow{\varepsilon}_+(\hat{n}) \equiv \hat{i} \otimes \hat{i} - \hat{j} \otimes \hat{j}$ and $\overleftrightarrow{\varepsilon}_\times(\hat{n}) \equiv \hat{i} \otimes \hat{j} + \hat{j} \otimes \hat{i}$. The rotation between these two basis systems defines the *polarization angle* ψ , which is measured counterclockwise from \hat{i} to $\hat{\ell}$, and relates the two polarization basis tensors as

$$\overleftrightarrow{e}_+ = \overleftrightarrow{\varepsilon}_+ \cos 2\psi + \overleftrightarrow{\varepsilon}_\times \sin 2\psi \quad (6.58)$$

$$\overleftrightarrow{e}_\times = -\overleftrightarrow{\varepsilon}_+ \sin 2\psi + \overleftrightarrow{\varepsilon}_\times \cos 2\psi. \quad (6.59)$$

Combining these expression, we can obtain the factored signal form $h^X(t; \mathcal{A}, \lambda) = \mathcal{A}^\mu h_\mu^X(t; \lambda)$ of Eq. (6.7), which was first derived in [102]. The four amplitudes $\{\mathcal{A}^\mu\}_{\mu=1}^4$ depend on the signal amplitude h_0 , the inclination angle ι via $A_+(h_0, \iota)$ and $A_\times(h_0, \iota)$ given in Eq. (6.6). They also depend on the polarization angle ψ , and the reference-time phase ϕ_0 , namely

$$\begin{aligned} \mathcal{A}^1 &= A_+ \cos \phi_0 \cos 2\psi - A_\times \sin \phi_0 \sin 2\psi, \\ \mathcal{A}^2 &= A_+ \cos \phi_0 \sin 2\psi + A_\times \sin \phi_0 \cos 2\psi, \\ \mathcal{A}^3 &= -A_+ \sin \phi_0 \cos 2\psi - A_\times \cos \phi_0 \sin 2\psi, \\ \mathcal{A}^4 &= -A_+ \sin \phi_0 \sin 2\psi + A_\times \cos \phi_0 \cos 2\psi, \end{aligned} \quad (6.60)$$

and the four (detector-dependent) basis functions $h_\mu^X(t; \lambda)$ are

$$\begin{aligned} h_1^X(t) &= a^X(t) \cos \phi(\tau^X(t)), \\ h_2^X(t) &= b^X(t) \cos \phi(\tau^X(t)), \\ h_3^X(t) &= a^X(t) \sin \phi(\tau^X(t)), \\ h_4^X(t) &= b^X(t) \sin \phi(\tau^X(t)), \end{aligned} \quad (6.61)$$

in terms of the *antenna-pattern functions* $a^X(t), b^X(t)$ given by the contractions

$$\begin{aligned} a^X(t; \hat{n}) &= \overleftrightarrow{d}^X(t) : \overleftrightarrow{\varepsilon}_+(\hat{n}), \\ b^X(t; \hat{n}) &= \overleftrightarrow{d}^X(t) : \overleftrightarrow{\varepsilon}_\times(\hat{n}). \end{aligned} \quad (6.62)$$

6. Fast and Accurate Sensitivity Estimation for Continuous-Gravitational-Wave Searches

Using the factored signal form of Eq. (6.7), the log-likelihood ratio Eq. (6.63) now takes the form

$$\ln \Lambda(x; \mathcal{A}, \lambda) = \mathcal{A}^\mu x_\mu - \frac{1}{2} \mathcal{A}^\mu \mathcal{M}_{\mu\nu} \mathcal{A}^\nu, \quad (6.63)$$

where we defined

$$x_\mu(\lambda) \equiv (x, h_\mu), \quad \text{and} \quad \mathcal{M}_{\mu\nu}(\lambda) \equiv (h_\mu, h_\nu), \quad (6.64)$$

in terms of the four basis function $h_\mu(t; \lambda)$ defined in Eq. (6.61). The 4×4 *antenna-pattern matrix* \mathcal{M} can be shown to be well approximated by the block-diagonal form

$$\mathcal{M} = \mathcal{S}^{-1} T_{\text{data}} \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \quad \text{with} \quad M \equiv \begin{pmatrix} A & C \\ C & B \end{pmatrix}, \quad (6.65)$$

defining the antenna-pattern coefficients A, B, C , which depend on the sky-position \hat{n} .

We see in Eq. (6.63) that the log-likelihood ratio is a quadratic function of the amplitudes \mathcal{A}^μ , and can therefore be analytically maximized [102] (or marginalized [108]) to yield the well-known \mathcal{F} -statistic:

$$\begin{aligned} \mathcal{F}(x; \lambda) &\equiv \max_{\mathcal{A}} \ln \Lambda(x; \mathcal{A}, \lambda) \\ &= \frac{1}{2} x_\mu \mathcal{M}^{\mu\nu} x_\nu, \end{aligned} \quad (6.66)$$

with $\mathcal{M}^{\mu\nu}$ defined as the inverse matrix to $\mathcal{M}_{\mu\nu}$ of Eq. (6.65).

6.10. Appendix C: Definition of the Geometric Factor R^2

The geometric factor R^2 can be explicitly expressed [106] as

$$R^2(\theta) = \frac{25}{4} [\alpha_1 A(\hat{n}) + \alpha_2 B(\hat{n}) + 2\alpha_3 C(\hat{n})], \quad (6.67)$$

with the sky-dependent antenna-pattern coefficients $\{A, B, C\}$ of Eq. (6.65), and

$$\alpha_1 \equiv \frac{1}{4} (1 + \cos^2 \iota)^2 \cos^2 2\psi + \cos^2 \iota \sin^2 2\psi, \quad (6.68)$$

$$\alpha_2 \equiv \frac{1}{4} (1 + \cos^2 \iota)^2 \sin^2 2\psi + \cos^2 \iota \cos^2 2\psi, \quad (6.69)$$

$$\alpha_3 \equiv \frac{1}{4} (1 - \cos^2 \iota)^2 \sin 2\psi \cos 2\psi. \quad (6.70)$$

One can show that R^2 averaged over $\psi \in [-\pi/4, \pi/4]$ and $\cos \iota \in [-1, 1]$ yields

$$\langle R^2 \rangle_{\cos \iota, \psi} = \frac{5}{2} (A(\hat{n}) + B(\hat{n})), \quad (6.71)$$

and further averaging \hat{n} isotropically over the sky yields

$$\langle R^2 \rangle_\theta = 1. \quad (6.72)$$

6.11. Appendix D: Distribution of \mathcal{F} -statistic Maximized over Correlated Templates

It has been a long-standing assumption (e.g. [29, 200]) that the distribution of the statistic $2\mathcal{F}^*(x) \equiv \max_{\lambda_i} 2\mathcal{F}(x; \lambda_i)$ in Gaussian noise x , maximized over a template bank $\lambda_i \in \mathbb{T}$ of

$i = 1 \dots \mathcal{N}$ (generally correlated) templates can be modelled by assuming maximization over an “effective” number of uncorrelated trials \mathcal{N}' instead, namely

$$P(2\mathcal{F}^* | \mathcal{N}') = \mathcal{N}' \text{cdf}_0(2\mathcal{F}^*)^{\mathcal{N}'-1} \text{pdf}_0(2\mathcal{F}^*), \quad (6.73)$$

where

$$\text{pdf}_0(2\mathcal{F}) = P(2\mathcal{F} | \rho = 0), \quad (6.74)$$

$$\text{cdf}_0(2\mathcal{F}) = \int_0^{2\mathcal{F}} \text{pdf}_0(2\mathcal{F}') \, d2\mathcal{F}', \quad (6.75)$$

where the (single-template) \mathcal{F} -statistic in pure Gaussian noise follows a central χ^2 distribution (with four degrees of freedom in the fully-coherent case Eq. (6.13), or $4N_{\text{seg}}$ degrees of freedom for a semi-coherent \mathcal{F} -statistic over N_{seg} segments, Eq. (6.22)).

We show here by counter-example that the model of Eq. (6.73) is not generally accurate, as correlations between templates do not simply modify \mathcal{N}' but also change the functional *form* of the distribution. It has been hypothesized previously [29] that these (already-observed) deviations might be due to certain approximations (c.f. [106]) used in the numerical implementation of the \mathcal{F} statistic. While such effects will account for some amount of deviation, one can show this effect to be quite small overall.

We demonstrate the fundamental statistical nature of this discrepancy by using a simpler example: We generate a time-series $\{x_j\}_{j=0}^{N-1}$ of $N = 200$ samples drawn from a Gaussian distribution and compute the Fourier transform \tilde{x}_k normalized to $E[|\tilde{x}_k|^2] = 2$, such that $2\mathcal{F}_2(x, f) \equiv |\tilde{x}(f)|^2$ follows a central χ^2 distribution with two degrees of freedom in every frequency bin f . We can therefore set $\text{pdf}_0(2\mathcal{F}_2) = \chi_2^2(2\mathcal{F}_2; 0)$ and use the corresponding cdf in Eq. (6.73).

We consider different cases of *oversampling* by zero-padding the time-series to a multiple (denoted as the oversampling factor in Fig. 6.10) of the original N time samples: the $N/2-1 = 99$ (positive) frequency bins without oversampling are strictly uncorrelated (and we also know that there can be at most $N = 200$ independent templates in total, given the length of the initial timeseries). With increasing oversampling, the correlations between frequency bins increase. We repeat this process 10^6 times for different noise realizations, and in each case we compute $2\mathcal{F}_2^*(x)$ over all the (positive) frequency bins of the Fourier power, and histogram these values. We then fit the number of effective templates \mathcal{N}' in the theoretical distribution of Eq. (6.73) by minimizing the (symmetric) Jensen–Shannon divergence between the measured and theoretical distributions. The results are shown in Fig. 6.10 for different cases of oversampling. We see that for increased oversampling, i.e. more correlations between ‘templates’ (i.e. frequency bins), the functional form of the histogram agrees less with the theoretical distribution assuming independent templates. The effect seems to saturate for oversampling $\gtrsim 10$, with $\mathcal{N}' \sim 230$ greater than the known maximal number (i.e. $N = 200$) of (strictly) independent templates in this vector space.

There is no simple or intuitive explanation for this effect that we are aware of, but it is reminiscent of a similarly surprising result found in the localization of the maximum over different assumed signal durations of transient CW signals, see Figs. 8 and 9 in [110]. The distribution of the statistic is identical in each time-step, but the steps are correlated, resulting in a peculiar non-uniform distribution of the location of the maximum.

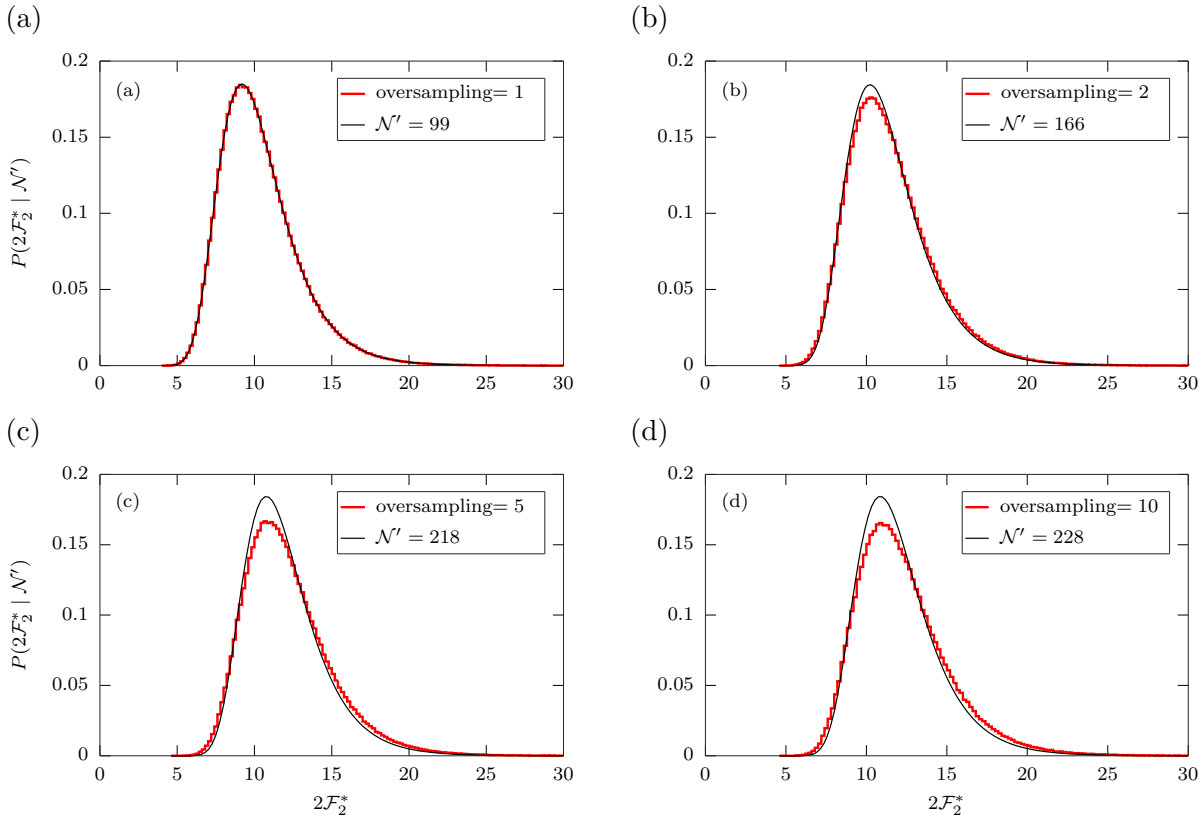


Figure 6.10.: *Stair-case plot*: histogram (over 10^6 repeated trials) of $2\mathcal{F}_2^* = \max_k |\tilde{x}_k|^2$ for Fourier transforms of Gaussian-noise timeseries, using different oversampling factors (a)–(d), where oversampling = 1 corresponds to the original FFT frequency resolution. *Solid thin line*: corresponding best-fit theoretical model Eq. (6.73) with an effective number of templates \mathcal{N}' .

7. Deep-Learning Continuous Gravitational Waves

This chapter is essentially a reprint of Dreissigacker et al. [30], with minor changes and reformatting as a chapter of this work.

7.0. Abstract

We present a first proof-of-principle study for using deep neural networks (DNNs) as a novel search method for continuous gravitational waves (CWs) from unknown spinning neutron stars. The sensitivity of current wide-parameter-space CW searches is limited by the available computing power, which makes neural networks an interesting alternative to investigate, as they are extremely fast once trained and have recently been shown to rival the sensitivity of matched filtering for black-hole merger signals [9, 10]. We train a convolutional neural network with residual (short-cut) connections and compare its detection power to that of a fully-coherent matched-filtering search using the WEAVE pipeline. As test benchmarks we consider two types of all-sky searches over the frequency range from 20 Hz to 1000 Hz: an “easy” search using $T = 10^5$ s of data, and a “harder” search using $T = 10^6$ s. Detection probability p_{det} is measured on a signal population for which matched filtering achieves $p_{\text{det}} = 90\%$ in Gaussian noise. In the easiest test case ($T = 10^5$ s at 20 Hz) the DNN achieves $p_{\text{det}} \sim 88\%$, corresponding to a loss in sensitivity depth of $\sim 5\%$ versus coherent matched filtering. However, at higher-frequencies and longer observation time the DNN detection power decreases, until $p_{\text{det}} \sim 13\%$ and a loss of $\sim 66\%$ in sensitivity depth in the hardest case ($T = 10^6$ s at 1000 Hz). We study the DNN generalization ability by testing on signals of different frequencies, spindowns and signal strengths than they were trained on. We observe excellent generalization: Only five networks, each trained at a different frequency, would be able to cover the whole frequency range of the search.

7.1. Introduction

Gravitational waves from binary mergers are now being observed routinely [1, 2, 237, 238] by the Advanced LIGO [44] and Virgo [46] detectors. In contrast, the much weaker and longer-lasting (days–months) narrow-band *continuous gravitational waves* (CWs) from spinning deformed neutron stars are yet to be detected, despite a multitude of searches over the past decade (see [85, 91, 177] for reviews) and continuing improvements in search methods (e.g. see [28] for a recent overview).

A key limitation of current search methods for CWs with unknown parameters is the “exploding computing cost problem”: give that a putative signal would be very weak, one needs to integrate as much data as possible in order to increase the signal-to-noise ratio (SNR). However, for a fully-coherent matched-filtering search (which is close to statistically optimal [108]), the corresponding computing cost grows as a high power $\sim T^n$ of the data timespan T , with $n \gtrsim 5$. This typically limits the longest coherent duration to days–weeks before the computing cost would become infeasible.

Therefore the class of semi-coherent methods has been developed, producing computationally cheaper searches. They allow the analysis of more data, typically resulting in better sensitivity than a corresponding coherent search at fixed computing cost (e.g. see [28, 180]). Such methods combined with massive amounts of computing power, either via local compute clusters or via

the distributed public computing platform Einstein@Home [7], currently yield the best state-of-the-art sensitivity to CW signals (e.g. see [239–241] for recent examples).

In this work we investigate *deep neural networks* (DNNs) [8, 152, 155] as a novel search method for CWs. The field of DNNs, also referred to as *Deep Learning*, has emerged as one of the most successful machine-learning paradigms in the last decade, dominating wide-ranging fields [8] such as image recognition, speech recognition and language translation, as well as certain board [242] and video games [243, 244].

More recently DNNs have started to draw attention in the field of gravitational-wave searches: (i) as a classifier for non-Gaussian detector transients (*glitches*) [21–24], (ii) as a search method for unmodelled burst signals [25, 26] in time-frequency images produced by coherent WaveBurst [27], and (iii) as a direct detection method for black-hole merger signals in gravitational-wave strain data [9–13].

This last approach (iii) is of particular interest to us, as [9, 10] have illustrated for the first time that DNNs can achieve a detection power comparable to that of (near-optimal) matched filtering, at a fraction of the search time. This is relevant for CW searches: while semi-coherent methods for wide-parameter-space searches are the most sensitive approach currently known, they are by design less sensitive than the statistical optimum achievable according to the Neyman-Pearson-Searle lemma [109].

With DNNs the computationally expensive step is shifted to the *preparation* stage of the search: the architecture tuning and “learning” of optimal network weights (i.e. the *training*), while the execution time on given input vectors is very short (typically fractions of a second). Determination of the noise-distribution (for estimation of the false alarm level p_{fa}) and measurement of upper limits require many repeated searches over different input data sets, with and without injected signals. The relative search speed advantage of DNNs compared to traditional search methods therefore accumulates dramatically over these operations allowing very fast and flexible search characterizations.

The plan of this chapter is as follows: In section 7.2 we define and characterize our test benchmarks. In section 7.3 we describe our deep-learning approach to searching for continuous gravitational waves: explaining the network architecture and how it was trained. In section 7.4 we characterize the performance our DNN achieves on the test benchmarks in comparison to the matched-filtering performance and how it generalizes beyond the benchmarks’ search parameters. And finally we discuss these results in section 7.5.

7.2. Comparison Test Benchmarks

7.2.1. Benchmark Definitions

In order to characterize the detection power of the DNN that we introduce in the next section, we define two benchmark search setups and measure the corresponding sensitivity achieved on them with a classical (near-optimal) matched-filter search method described in Sec. 7.2.2.

We compare the sensitivity in the Neyman-Pearson sense, also known as the receiver-operator characteristic (ROC), using the “upper limit” conventions used in most CW searches (cf. [28]): measure the detection probability p_{det} at a chosen false-alarm level p_{fa} for a signal population of fixed amplitude h_0 , with all other signal parameters (i.e. polarization, sky-position, frequency and spindown) drawn randomly from their priors. In order to characterize the signal strength in noise, we use *sensitivity depth* \mathcal{D} [28, 199], defined as

$$\mathcal{D} \equiv \frac{\sqrt{\mathcal{S}}}{h_0}, \tag{7.1}$$

where \mathcal{S} is the power-spectral density (PSD) of the (Gaussian) noise at the signal frequency, and h_0 is the signal amplitude. In particular we are interested in the sensitivity depth $\mathcal{D}^{90\%}$ that

corresponds to the signal amplitude $h_0^{90\%}$ at which the search yields a detection probability of $p_{\text{det}} = 90\%$ at a fixed false-alarm level, which here is chosen as $p_{\text{fa}} = 1\%$ per 50 mHz frequency band.

We consider two *all-sky* searches (parameters summarized in Table. 7.1) over a range in frequency f and first-order spindown \dot{f} , one using $T = 10^5$ s \sim 1.2 days, and one using $T = 10^6$ s \sim 12 days of data assuming a single detector (chosen as LIGO Hanford). These two searches could

parameter name	parameter values
data span	$T = 10^5$ s / $T = 10^6$ s
detectors	LIGO Hanford
noise	stationary, white, Gaussian
sky-region	all-sky
frequency band	$f \in [20, 1000]$ Hz
spin-down range	$\dot{f} \in [-10^{-10}, 0]$ Hz/s

Table 7.1.: Definition of the two benchmark searches.

realistically be performed with coherent matched filtering. The required computing cost for the search and its characterization (upper limits, false alarm level) however would still require a large cluster of, say, $\mathcal{O}(1000)$ cores for over a month or so (see Table 7.2). Therefore actually performing these two full searches only for the purpose of characterizing the matched-filtering sensitivity would be infeasible. Instead we characterize the matched-filter search on only five narrow frequency bands of width $\Delta f = 50$ mHz starting at frequencies $f_0 = 20, 100, 200, 500$ and 1000 Hz, yielding a total of ten representative test cases.

7.2.2. Weave Matched-filtering Sensitivity

For the matched-filter search we use the recently-developed WEAVE code [145], which implements a state-of-the-art CW search algorithm [245] based on summing coherent \mathcal{F} -statistics [102, 197] over semi-coherent segments on optimal lattice-based template banks [246, 247]. This code can also perform fully-coherent (i.e. single-segment) \mathcal{F} -statistic searches, which we use for the present proof-of-principle study. The benchmark search definitions in Table 7.1 are chosen in such a way that a fully-coherent search is still computationally feasible. This yields a simpler and cleaner comparison than using a semi-coherent search setup, as we can easily design near-optimal search setups (by using relatively fine template banks) without the extra complication of requiring costly sensitivity-optimization at fixed computing cost [180, 191, 245].

The WEAVE template banks are characterized by a maximal-mismatch parameter m , which controls how fine the templates are spaced in parameter space. These are chosen as $m = 0.1$ and $m = 0.2$ for the two searches with $T = 10^5$ s and $T = 10^6$ s, respectively. The reason for choosing the larger mismatch value (i.e. coarser template bank) in the $T = 10^6$ s case is to keep the computing cost of the corresponding test-cases still practically manageable, as the coherent cost scales with mismatch parameter as $\propto m^2$ for a four-dimensional template bank (e.g. see Eq.(24) in [246]).

By repeated injections of signals in the data and recovery of the loudest \mathcal{F} -statistic candidate in the template bank, one can measure the relative SNR-loss μ compared to a perfectly-matched template. The resulting measured average mismatch $\langle \mu \rangle$ quantifies in some sense how close to “optimal” the matched-filter sensitivity will be (compared to an infinite-computing cost search with $m = 0$), and is found as $\langle \mu \rangle \sim 5\%$ and $\langle \mu \rangle \sim 11\%$, respectively for the two searches.

Using the template-counting and timing models [145, 205, 248] for WEAVE and the resampling \mathcal{F} -statistic, we can estimate the total number of templates and the corresponding total runtime for these two benchmark searches as ~ 78 days and $\sim 45\,000$ days on a single CPU core, re-

7. Deep-Learning Continuous Gravitational Waves

spectively. Table. 7.2 provides a summary of the WEAVE search parameters and characteristics.

name	$T = 10^5$ s	$T = 10^6$ s
mismatch parameter m	0.1	0.2
average SNR loss $\langle \mu \rangle$	5%	11%
Number of templates \mathcal{N}	$4 \cdot 10^{11}$	$3 \cdot 10^{14}$
Search time [single CPU core]	$6.7 \cdot 10^6$ s	$3.9 \cdot 10^9$ s

Table 7.2.: WEAVE parameters and characteristics for the two searches.

In order to estimate the sensitivity for the ten test cases defined in the previous section (i.e. five frequency slices of $\Delta f = 50$ mHz for each search of $T = 10^5$ s and $T = 10^6$ s, respectively), we first determine the corresponding detection thresholds \mathcal{F}_{th} on the \mathcal{F} -statistic corresponding to a false-alarm level of $p_{\text{fa}} = 1\%$ for each case. This is done by repeatedly (10^5 times for $T = 10^5$ s, and $\sim 10^4$ times for $T = 10^6$ s, respectively) performing each search over Gaussian noise and thereby recording the distribution of the loudest candidate, which yields the relationship between threshold and false-alarm level. The corresponding detection probability p_{det} for any given signal population of fixed \mathcal{D} is obtained by injecting signals into Gaussian noise data and measuring how many times the loudest candidate exceeds the detection threshold. By varying the injected \mathcal{D} we can eventually find $\mathcal{D}^{90\%}$ for the desired $p_{\text{det}} = 90\%$ (e.g. see [28] for more details and discussion of this standard “upper limit” procedure). By a final injection+recovery Monte-Carlo we can verify that the achieved WEAVE detection probability for the quoted thresholds and signal strengths $\mathcal{D}^{90\%}$ in Table. 7.3 is $p_{\text{det}} \sim 90\% - 91\%$, which is sufficiently accurate for our present purposes.

The sky template resolution grows as $\propto f^2$ as a function of frequency f , resulting in a corresponding increase in the number of templates at higher frequency. This increases the number of “trials” in noise at the higher-frequency slices, which results in a corresponding increased false-alarm threshold (chosen in order to keep the false-alarm level at $p_{\text{fa}} = 1\%$) as well as an increased computing cost, shown in Table. 7.3.

f_0	20 Hz	100 Hz	200 Hz	500 Hz	1000 Hz
$T = 10^5$ s					
$\mathcal{N}_{\Delta f}$	$5 \cdot 10^5$	$1 \cdot 10^7$	$5 \cdot 10^7$	$3 \cdot 10^8$	$1 \cdot 10^9$
CPU $_{\Delta f}$ [s]	0.1	4.9	19	$2.3 \cdot 10^2$	$1.7 \cdot 10^3$
$\mathcal{F}_{\text{th}}(p_{\text{fa}})$	20.6	23.6	25.1	27.0	28.6
$T = 10^6$ s					
$\mathcal{N}_{\Delta f}$	$3 \cdot 10^8$	$8 \cdot 10^9$	$3 \cdot 10^{10}$	$2 \cdot 10^{11}$	$8 \cdot 10^{11}$
CPU $_{\Delta f}$ [s]	45	$3 \cdot 10^3$	$1.4 \cdot 10^4$	$1.6 \cdot 10^5$	$6.9 \cdot 10^5$
$\mathcal{F}_{\text{th}}(p_{\text{fa}})$	27.5	31.1	32.5	34.2	36.2

Table 7.3.: WEAVE characteristics for the ten test cases, each covering a frequency “slice” of $\Delta f = 50$ mHz, starting at f_0 , of the full searches defined in Table. 7.1. The detection thresholds \mathcal{F}_{th} correspond to a false-alarm level of $p_{\text{fa}} = 1\%$ over the band Δf . $\mathcal{N}_{\Delta f}$ is the number of templates used per respective frequency band. CPU $_{\Delta f}$ denotes the search time in seconds for the respective Δf band on a single CPU core.

$\mathcal{D}^{90\%} [\text{Hz}^{-1/2}]$	$f_0 = 20 \text{ Hz}$	100 Hz	200 Hz	500 Hz	1000 Hz
$T = 10^5 \text{ s}$	11.4	10.8	10.4	9.9	9.7
$T = 10^6 \text{ s}$	29.3	28.2	27.6	26.8	26.0

Table 7.4.: Measured WEAVE “upper limit” sensitivity $\mathcal{D}^{90\%}$ at false-alarm level of $p_{\text{fa}} = 1\%$.

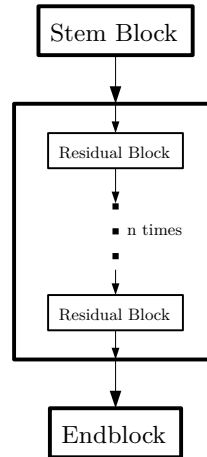


Figure 7.1.: Illustration of the general network architecture used in this study.

7.3. Deep-Learning CWs

Our general approach is similar to that of [9, 10] in that we directly use the detector strain data as our network input, and train a simple classifier with two output neurons for the classes “noise” and “signal (in noise)”. However, given that CW signals are long in duration and narrow in frequency, instead of using the time-series input it makes more sense in our case to use the frequency-domain representation of that data. We therefore provide the real- and imaginary parts of the fast Fourier transform (FFT) of the data as a two-dimensional input vector over frequency bins, using the native FFT resolution of $1/T$. We chose the network input size to be sufficiently large to contain the widest signal (signals get stretched in frequency domain by spindown \dot{f} and Doppler shifts) *twice*, so that we can slide the network along the frequency axis in steps of half the network input width, guaranteeing that one input window will always contain the full signal.

7.3.1. Network Architecture

We started experimenting with DNN architectures similar to those described in [9, 10], but eventually by trial and error converged on a ResNet architecture [170], which showed better performance for our problem cases.

We have chosen slightly different networks for the two searches ($T = 10^5 \text{ s}$ and $T = 10^6 \text{ s}$) of Table. 7.1, as these correspond to signals with rather different width in frequency domain: the network in the $T = 10^5 \text{ s}$ cases contains six instances of a residual block, while in the $T = 10^6 \text{ s}$ cases the network uses twelve.

The network layers can be separated into three parts: the stem block, a block of multiple residual blocks, and an end-block, see Fig 7.1. The stem block consists of a standard convolutional layer, while each of the residual blocks is built according to [170]. The endblock contains a dense softmax layer with two final output neurons, corresponding to the estimated probability p_{signal} that the input contains a signal, and $p_{\text{noise}} = 1 - p_{\text{signal}}$ for pure noise sample. The DNNs are implemented in the Keras framework[151] on top of a Tensorflow[249] backend.

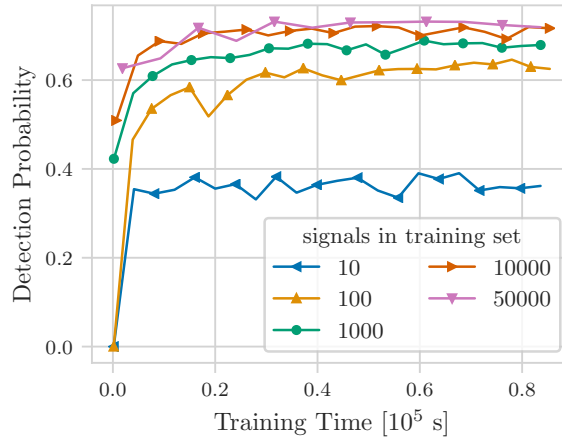


Figure 7.2.: Validation detection probability for $T = 10^5$ s, $f_0 = 1000$ Hz, “H1” for training with training sets containing 10, 100, 1000, 10000 and 50000 signals.

7.3.2. DNN Training and Validation

Training the network is performed on a synthesized data-set of input vectors, where half contain pure Gaussian noise, and half contain a signal added to the noise. One full pass through this training set is commonly referred to as a training *epoch*. Using a pre-computed set of 10 000 signals, each signal is added to 24 dynamically-generated noise realizations, which are also used as pure-noise inputs. The number of signals in the training set was determined empirically, as using more signals gives diminishing performance improvements (see Fig. 7.2). The signals are scaled to a fixed depth $\mathcal{D}_{\text{training}}^{90\%}$ for each test case and randomly shifted in frequency within the network input window. These training depths were estimated semi-analytically using the method of [28, 205], and differ slightly from the final measured values $\mathcal{D}^{90\%}$ of Table. 7.4, which had not yet been available at the time of training. When testing the network on signals of different depths, the detection probability behaves as expected, see Sec. 7.4.4. Furthermore, we found that using a different choice of training depth did not significantly affect training success.

Every few epochs of training, we perform a *validation* step, where the detection probability of the network is measured on an independent data set. This validation set contains another 20 000 input vectors, half containing signals in noise (of fixed depth $\mathcal{D}^{90\%}$), and half containing noise only.

In order to compute the network’s detection probability $p_{\text{det}}^{\text{DNN}}$, we treat the output neuron p_{signal} as a statistic, and follow the usual “upper limit” procedure described in Sec. 7.2.2: we repeatedly run the network on Gaussian noise inputs in order to determine the $p_{\text{fa}} = 1\%$ detection threshold. We then run the network on the signal set and measure for what fraction of signals the statistic exceeds that threshold.

The evolution of the detection probability as a function of training epoch (or similarly, as a function of training time) is presented in Fig. 7.3, illustrating the progress of learning. In order to test the variability and dependence of the learning success on the random initialization of the network, we train a “cloud” of ~ 100 differently-initialized network instances. We use the network at its point of best validation performance from each test case for the further test results presented in the next sections.

Most of the training was performed on Nvidia GTX 750 GPUs. We see in Fig. 7.3 that for most cases the improvements in detection probability seem to have leveled off after the training time (about one day in the $T = 10^5$ s cases, and about 10 days in the $T = 10^6$ s cases). However, in the case of $T = 10^6$ s, $f_0 = 1000$ Hz, “H1” seen in Fig.7.3d (and also for $T = 10^6$ s, $f_0 = 500$ Hz, “H1”, not shown), there still seems to be a slowly increasing trend in detection probability at the end

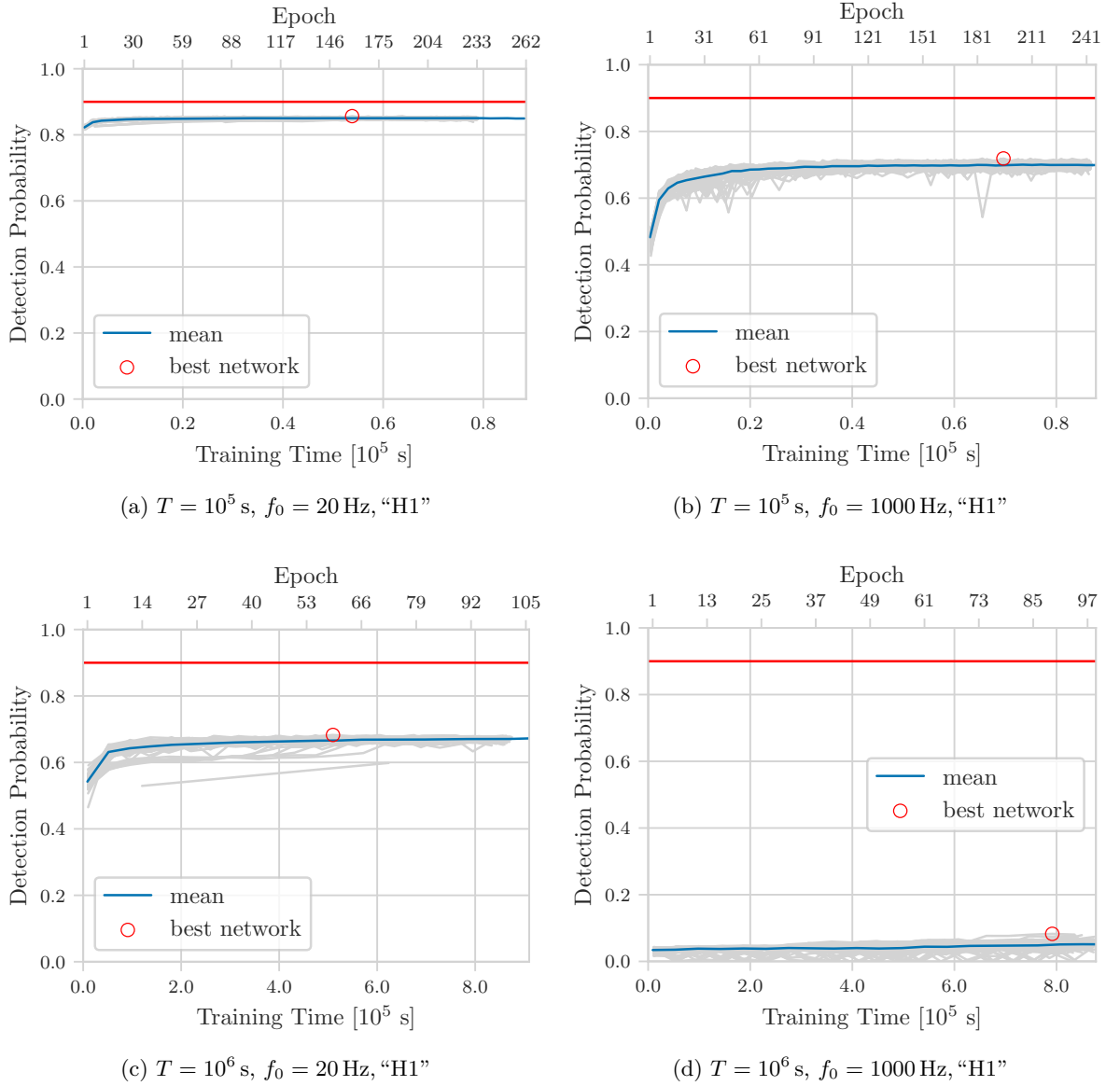


Figure 7.3.: Validation detection probability $p_{\text{det}}^{\text{DNN}}$ of the DNN versus training time (or mean trained epoch) for 100 different network instances trained for each of four test cases: **(a)** $T = 10^5$ s, $f_0 = 20$ Hz, "H1", **(b)** $T = 10^5$ s, $f_0 = 1000$ Hz, "H1", **(c)** $T = 10^6$ s, $f_0 = 20$ Hz, "H1" and **(d)** $T = 10^6$ s, $f_0 = 1000$ Hz, "H1", all trained on Nvidia GTX 750. The solid horizontal line denotes the matched-filtering detection performance of $p_{\text{det}} = 90\%$.

7. Deep-Learning Continuous Gravitational Waves

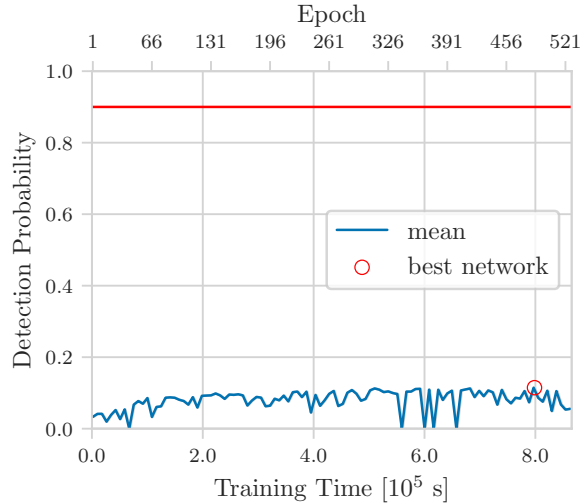


Figure 7.4.: Validation detection probability $p_{\text{det}}^{\text{DNN}}$ of the DNN versus training time for a single network trained on Nvidia TITAN V for the case $T = 10^6$ s, $f_0 = 1000$ Hz, “H1”.

of training time. Therefore we trained a single network instance for these two cases again on a more powerful Nvidia TITAN V GPU for many more epochs, until the validation detection probability seemed to level off, which is shown in Fig. 7.4.

Overall we observe an dramatic increase in “difficulty” the DNN has in learning the different test cases along the direction of increasing data span T and frequency f , also seen clearly in Table. 7.5. In the easiest case of $T = 10^5$ s, $f_0 = 20$ Hz, “H1” the DNN achieves a detection probability of $p_{\text{det}}^{\text{DNN}} \sim 88\%$, nearly rivalling matched-filtering performance, while in the hardest case of $T = 10^6$ s, $f_0 = 1000$ Hz, “H1” it only manages $p_{\text{det}} \sim 13\%$ (also see Table 7.5). This may not be very surprising, given that the cases become increasingly more compute-intensive (more templates) along the same axis for matched filtering, as seen in Table. 7.3. In the frequency-domain input vectors of the DNN, this would manifest by the signals being more widely spread-out due to increased frequency drift $\dot{f}T$ and Doppler stretching.

7.4. Testing DNN Performance

After the training and validation steps, we perform a series of tests on the best DNN found for each test case (i.e. with the highest $p_{\text{det}}^{\text{DNN}}$ over all validation steps), in order to more fully characterize its performance as a CW detection method. In these tests we simulate the signals and noise directly for any given depth using the standard CWLALSUITE[250] machinery, in order to independently verify the network performance. Hence we are not using a traditionally fixed *testing set* but generate it on demand.

7.4.1. Verifying Detection Probabilities

As a sanity check we measure again the detection probability $p_{\text{det}}^{\text{DNN}}$ at $p_{\text{fa}} = 1\%$ for the ten cases over the respective frequency bands for a signal population at the matched-filtering $\mathcal{D}^{90\%}$ of Table. 7.4. The resulting DNN test results obtained with the independent test-pipeline are given in Table. 7.5. These results usually agree to ~ 2 percentage points in detection probability with the corresponding best $p_{\text{det}}^{\text{DNN}}$ originally observed in the validation step, seen in Figs. 7.3,7.4.

A second interesting question is how the detection probability depends on the false-alarm level p_{fa} (commonly referred to as ROC curve) for a fixed signal population. This is shown in Fig. 7.5 in comparison to the matched-filter ROC.

$p_{\text{det}}^{\text{DNN}} [\%]$	$f_0 = 20 \text{ Hz}$	100 Hz	200 Hz	500 Hz	1000 Hz
$T = 10^5 \text{ s}$	$87.6^{+0.7}_{-0.6}$	$85.4^{+0.7}_{-0.7}$	$84.1^{+0.7}_{-0.7}$	$80.2^{+0.8}_{-0.8}$	$73.0^{+0.9}_{-0.9}$
$T = 10^6 \text{ s}$	$68.8^{+0.9}_{-0.9}$	$50.0^{+1.0}_{-1.0}$	$38.7^{+0.9}_{-1.0}$	$25.4^{+0.8}_{-0.9}$	$13.1^{+0.6}_{-0.7}$

Table 7.5.: Detection probabilities in % of the best networks for each case at false alarm level $p_{\text{fa}} = 1\%$ and 90% matched-filtering depth.

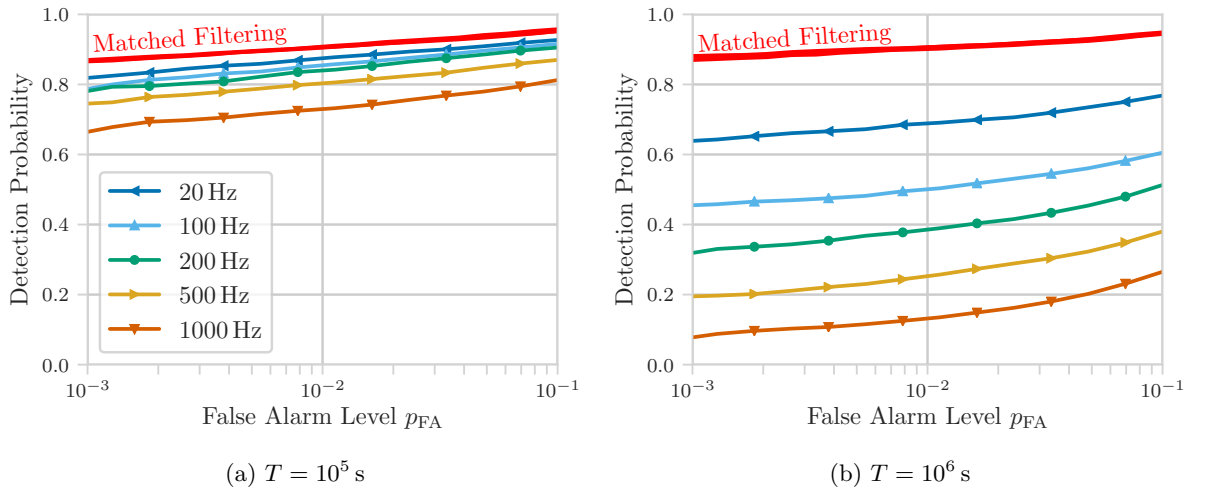


Figure 7.5.: ROC-curve: Detection probability p_{det} versus p_{fa} for the 10^5 s search (left) and the 10^6 s search (right). The solid red lines indicate the measured ROC curves for matched filtering.

7.4.2. Generalization in Frequency f

If we want to perform a search over the whole frequency range (e.g. as defined in Table. 7.1) using DNNs, we would need to determine how many different networks we have to train in order to cover this range with a reasonable overall sensitivity. Alternatively we can also train a single DNN with signals drawn from the full frequency range of the search and compare its performance.

The results of these tests are shown in Fig. 7.6, which show how the five DNNs, trained at their respective frequencies f_0 , perform over the full frequency range of the search. In addition we show the performance of another network that has been trained directly over the full frequency range.

We see that the “specific” networks trained only on a narrow frequency range still perform reasonably well over a fairly broad range of frequencies, and especially that networks trained at higher frequencies generalize well to lower frequencies. This result shows that a small number of networks $\mathcal{O}(5)$ would be able to cover the whole frequency range at a similar detection performance that was obtained on the individual training frequencies. Furthermore, it seems quite feasible to train a single network over the full frequency range directly, achieving similar (albeit slightly lower) performance to the “specialized” networks trained at narrow frequency bands.

7.4.3. Generalization in Spin-down \dot{f}

A further interesting aspect to consider is how far in spindown \dot{f} the performance network extends beyond the range that it was trained on, i.e. $\dot{f} \in [-10^{-10}, 0] \text{ Hz/s}$ as given in Table. 7.1.

7. Deep-Learning Continuous Gravitational Waves

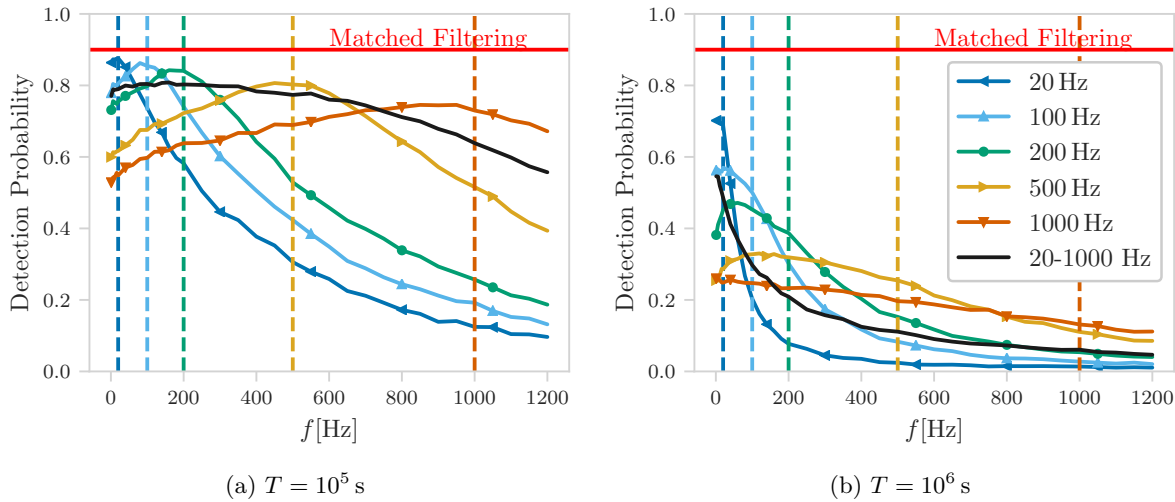


Figure 7.6.: Detection probability p_{det} versus injection frequency f for networks trained at five different frequencies and for a network trained with signals drawn from the full frequency range (solid black line). The dashed vertical lines mark the respective training frequencies for the five “specialized” networks. The horizontal dashed line represents the coherent matched filtering detection performance.

This is shown in Fig. 7.7. We see that the DNN detection probability remains high even for spindowns that are 1-2 orders of magnitude larger than the training range. In particular, networks trained at higher frequencies seem to have a wider generalization range in spindown, which makes sense as they would have learned to recognize signal shapes with larger Doppler broadening, a qualitatively similar effect to having more spindown.

7.4.4. Generalization in Signal Strength

Another important issue is how well the DNN generalizes for signals of different strength \mathcal{D} , given that we only trained each network at one specific depth $\mathcal{D}_{\text{training}}^{90\%}$, an estimate of the matched filtering depth. The results of this test are shown in the efficiency plots of Fig. 7.8. We see that generally the dependence of $p_{\text{det}}(\mathcal{D})$ for the DNNs seems to be quite similar to that of matched filtering, but shifted to its overall (lower) performance level.

Conversely we also calculated the “upper limit” sensitivity depth $\mathcal{D}_{\text{DNN}}^{90\%}$ where the network achieves 90% detection probability (see Table 7.6). These values correspond to a sensitivity loss of 5% – 21% (as a function of frequency) for the $T = 10^5$ s search, and 26% – 66% for the $T = 10^6$ s search.

$\mathcal{D}_{\text{DNN}}^{90\%} [\text{Hz}^{-1/2}]$	$f_0 = 20$ Hz	100 Hz	200 Hz	500 Hz	1000 Hz
$T = 10^5$ s	10.8	10.0	9.5	8.6	7.7
$T = 10^6$ s	21.6	16.5	14.3	11.1	8.9

Table 7.6.: Sensitivity depths $\mathcal{D}_{\text{DNN}}^{90\%}$ at false-alarm level of $p_{\text{fa}} = 1\%$ achieved by the network for the ten test cases. The respective matched filter depths can be found in Table 7.4.

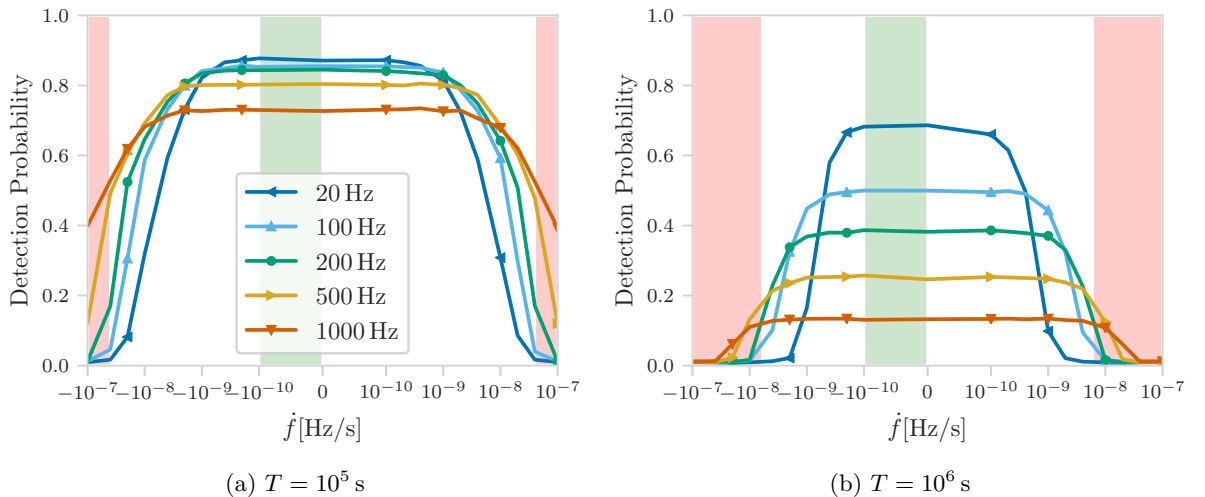


Figure 7.7.: Detection probability p_{det} versus injected spin-down \dot{f} for networks trained at give different frequencies. The green shade in the middle marks the 10^{-10} Hz/s wide spin-down band the networks were trained on. The x-axis is plotted as a symmetric logarithm, i.e. logarithmical for the larger negative values, linear for $|\dot{f}| < -10^{-10}$ Hz/s and logarithmical for the larger positive values. The red shades at the edges illustrates where we start losing SNR purely by the network input window being smaller than the widest signals.

7.4.5. Timing

The total amount of computational resources needed, is another interesting point of comparison to a matched filter search. The total search times for using the matched-filter WEAVE method on the two benchmark searches can be found in Table 7.2.

For the DNN the total computation time consists of two parts: Training time and *prediction* time (i.e. calculating one output statistic p_{signal} for one input data vector). The training time for the two network architectures is ~ 1 d and ~ 10 d per network for the $T = 10^5$ s and $T = 10^6$ s cases, respectively. Only part of this time is actually spent on training the network, another part is calculating the detection probability of the network every few epochs in order to monitor the progress of training.

The prediction time in comparison is almost negligible. The smaller networks for the $T = 10^5$ s cases require ~ 3 ms for processing one input window. The larger networks for the $T = 10^6$ s cases need ~ 10 ms per prediction. Each search requires a different number of sliding input windows to cover the whole frequency range, and the total search time can be found in table 7.7.

Cost [s]	Training	Search	Follow-up	Total
$T = 10^5$ s	$4.3 \cdot 10^5$	58.8	$2.2 \cdot 10^4$	$4.5 \cdot 10^5$
$T = 10^6$ s	$4.3 \cdot 10^6$	196	$6.5 \cdot 10^7$	$6.9 \cdot 10^7$

Table 7.7.: DNN computing cost (in seconds) for training, search and follow-up (using matched-filtering). The respective matched-filtering cost can be found in Table 7.2

An important detail to note in a direct comparison between matched filtering and a pure classifier “signal” vs “noise” DNN search is that matched filtering yields far more information on outlier candidates that cross the threshold. In particular, its signal parameters will be well

7. Deep-Learning Continuous Gravitational Waves

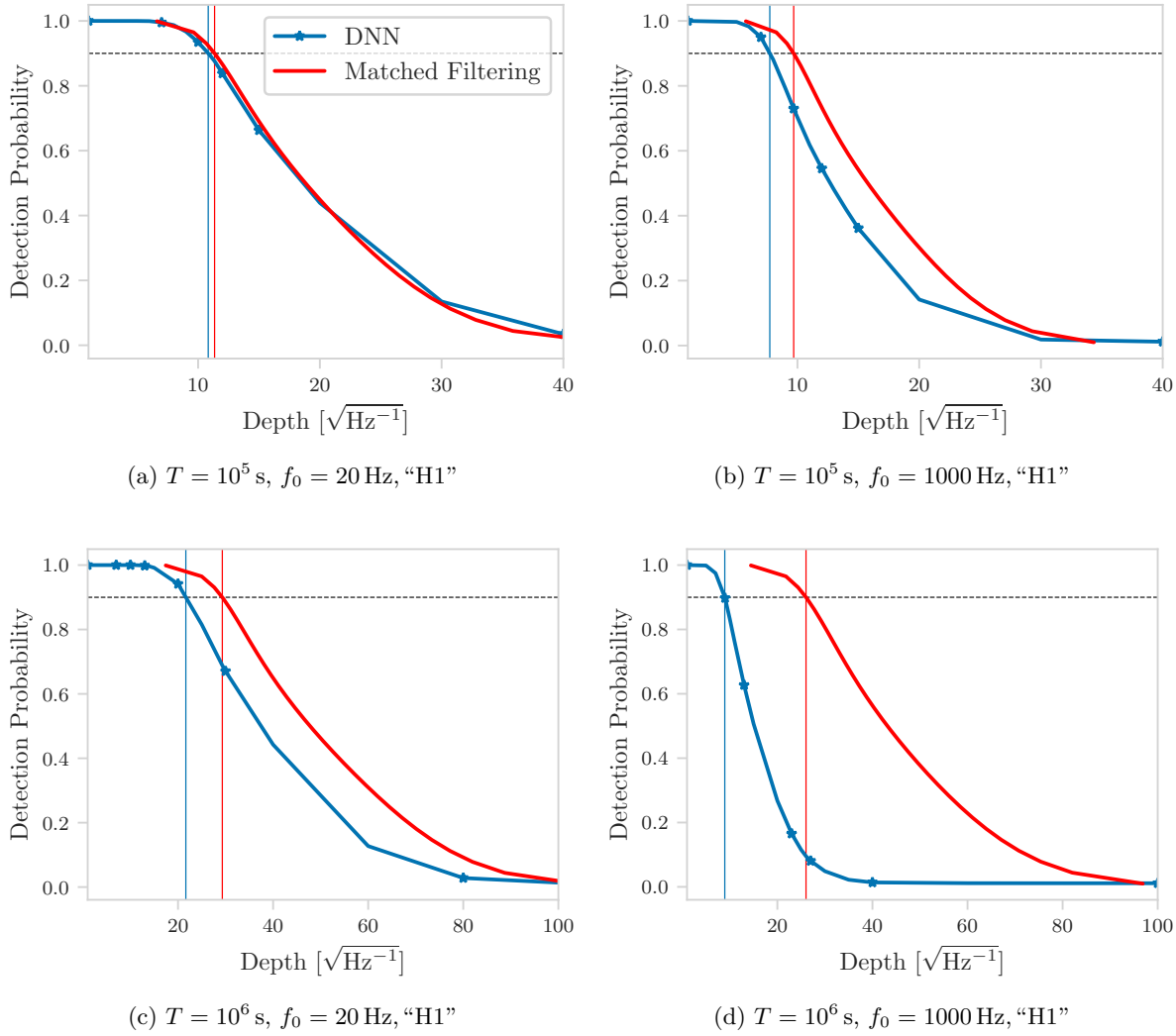


Figure 7.8.: Detection probability p_{det} versus injection depth \mathcal{D} for networks trained on the respective matched-filtering depth $\mathcal{D}^{90\%}$ (indicated by the vertical red line with the at 90%). The blue vertical line gives the sensitivity depth for the DNN at 90% detection probability.

constrained already, allowing a follow-up search to be performed in a small region of parameter space. The DNN classifier, on the other hand, would flag input windows (of width Δf_{IW}) in frequency as outliers to be followed up. Assuming we follow up two input windows per candidate, one can estimate the total expected follow-up cost (using matched-filtering) as a fraction $2(\Delta f_{\text{IW}}/\Delta f) p_{\text{fa}}$ of the total matched-filtering cost (see Table 7.2), where $p_{\text{fa}} = 1\%$ is the false-alarm probability per $\Delta f = 50$ mHz band.

Therefore even including all the training time and assuming a matched-filter follow-up, the DNN search would still seem to be requiring less computing power. At the present stage, however, we cannot realise this potential benefit given that our DNN search so far is far less sensitive overall.

7.5. Discussion

In this work we have shown that Deep Learning (DNN) can in principle be used to directly search for CW signals in data, at substantially faster search times than matched filtering. For the hand-optimized network architecture studied here, the DNN detection probability (at fixed false alarm) is found to be somewhat competitive (88% – 73% over the full frequency range) with matched filtering (90%) for short data-spans of $T \sim 1$ day, while the detection performance falls short (69% – 13%) for a longer data span of $T \sim 12$ days. On the plus side, the DNN search shows a surprising ability to extend further in frequency and spindown than it was trained for, and is generally much faster in search performance than matched filtering.

Overall we find that Deep Learning has potential to become a useful CW search tool, but probably a lot more work and effort is required to achieve this. A few immediate ideas we are planning to pursue next in this project are: automated large-scale architecture optimization, training for parameter-estimation in addition to pure classification, extending it to a multi-detector search, and investigating performance on non-Gaussian detector noise.

8. Deep-Learning Continuous Gravitational Waves: Multiple detectors and realistic noise

This chapter is essentially a reprint of Dreissigacker and Prix [31], with minor changes and reformatting as a chapter of this work.

8.0. Abstract

The sensitivity of wide-parameter-space searches for continuous gravitational waves is limited by computational cost. Recently it was shown that Deep Neural Networks (DNNs) can perform all-sky searches directly on (single-detector) strain data [30], potentially providing a low-computing-cost search method that could lead to a better overall sensitivity. Here we expand on this study in two respects: (i) using (simulated) strain data from two detectors simultaneously, and (ii) training for directed (i.e. single sky-position) searches in addition to all-sky searches. For a data timespan of $T = 10^5$ s, the all-sky two-detector DNN is about 7% less sensitive (in amplitude h_0) at low frequency ($f = 20$ Hz), and about 51% less sensitive at high frequency ($f = 1000$ Hz) compared to fully-coherent matched-filtering (using WEAVE). In the directed case the sensitivity gap compared to matched-filtering ranges from about 7–14% at $f = 20$ Hz to about 37–49% at $f = 1500$ Hz. Furthermore we assess the DNN’s ability to generalize in signal frequency, spindown and sky-position, and we test its robustness to realistic data conditions, namely gaps in the data and using real LIGO detector noise. We find that the DNN performance is not adversely affected by gaps in the test data or by using a relatively undisturbed band of LIGO detector data instead of Gaussian noise. However, when using a more disturbed LIGO band for the tests, the DNN’s detection performance is substantially degraded due to the increase in false alarms, as expected.

8.1. Introduction

Observing gravitational waves from compact binary mergers has become routine [1, 2, 237, 238, 251]. The long-lasting but weak narrow-band signals from spinning non-axisymmetric neutron stars called continuous gravitational waves (CWs) however remain elusive at the current sensitivity of the Advanced LIGO [44] and Virgo [46] detectors. Despite great improvements in the search methods (see e.g. [28] for a recent review) and numerous searches conducted on past and recent detector data (see Refs. [85, 91, 177] for reviews), no CW discovery has been made yet.

The sensitivity of CW searches is typically limited by the prohibitive computing cost. A CW signal is expected to last longer than the observation time. Hence, to increase the signal-to-noise ratio (SNR) of a search, it needs to integrate over as much data as possible but for a typical fully coherent matched-filtering search the computing cost grows as a high power $\sim T^n$, $n \geq 5$ of the time span of data T . Therefore these statistically almost optimal searches [108] can only be performed with coherence times of days to weeks at most.

The main method to circumvent this limitation is to use semicoherent methods. These consist of using shorter coherent segments and combining their results incoherently resulting in an improved sensitivity at fixed computing cost [126, 180]. Nevertheless the currently most sensitive wide parameter space searches (see e.g. [252–254]) are using massive amounts of computational

resources, either in the form of local computer clusters or the distributed computing project Einstein@Home [7].

In this work we study the feasibility of *deep neural networks* (DNNs) as an alternative search method. DNNs have been shown to be able to approximate any Borel-measurable function [153, 154] (see also [155] for a more general discussion). Therefore they should in principle be able to approximate gravitational-wave-search methods.

In fact the method of training a DNN, also called *deep learning*, has been established to be able to detect gravitational waves directly from strain data [9–15] for signals from mergers of compact objects. More recently it was used for the first time on simulated continuous gravitational wave signals [30] and it was applied to the related long transient signals [16]. Furthermore DNNs have been studied as a follow-up method for CW searches [17, 18], as well as for parameter estimation of searches for compact binary merger signals [19, 20] and for a multitude of other gravitational-wave-search related applications such as classifying disturbances (*glitches*) and searches for unmodeled burst signals [21–27].

In this work we continue the effort towards building a competitive neural-network-based search method for CWs by gradually moving towards more realistic test- and training scenarios: by simultaneously using data from two detectors, by including directed search cases, and by testing the trained DNNs on Gaussian data with gaps and on real LIGO data with varying degrees of instrumental disturbances.

The plan of this chapter is as follows: in Sec 8.2 we define the new benchmark cases, we discuss the updated deep-learning approach in Sec. 8.3, we characterize the performance of the DNNs on the benchmarks by testing them on Gaussian noise in Sec. 8.4 and finally we extend this characterization to the intricacies of real detector noise in Sec. 8.5. In Sec. 8.6 we discuss our results.

8.2. Comparison Test Benchmarks

We characterize the DNNs as search method on the following two-detector benchmark searches, each assuming two different timespan baselines of $T = 10^5$ s and $T = 10^6$ s: an all-sky search and two directed searches pointing at the supernova remnants of Cassiopeia A (CasA) and G347.3-0.5 (G347), respectively. For the coherent matched-filter comparison we use the WEAVE search code [145] in the same way as in [30].

We measure the sensitivity of the DNNs and the matched-filtering searches by determining the detection probability p_{det} at a chosen false-alarm level of $p_{\text{fa}} = 1\%$ per 50 mHz frequency band. The false-alarm level corresponds to a threshold on the respective detection statistic of the DNNs and the matched-filtering searches for a signal population of fixed signal amplitude given in terms of the sensitivity depth \mathcal{D} [28, 199], defined as

$$\mathcal{D} \equiv \frac{\sqrt{\mathcal{S}}}{h_0}, \quad (8.1)$$

where \mathcal{S} is the power spectral density of the noise at the signal frequency, and h_0 is the signal amplitude. In particular we will use $\mathcal{D}^{90\%}$ to refer to the 90%-upper limit depth, corresponding to a signal amplitude $h^{90\%}$ where a search method achieves a detection probability of $p_{\text{det}} = 90\%$ at a false-alarm threshold of $p_{\text{fa}} = 1\%$ per 50 mHz frequency band.

For reasons of speed and simplicity, at this stage of the project we still use simulated Gaussian noise for the DNN training and for the matched-filtering comparison. However, in Sec. 8.5 we do show tests of our DNN search pipeline on data with realistic gaps and also using real LIGO detector data.

The two-detector benchmarks are similar to the previously-used single-detector benchmarks of [30], as they encompass data spans of $T = 10^5$ s and $T = 10^6$ s, and the all-sky searches cover the same parameter space as the previous single-detector all-sky cases (see Table 8.1).

The new directed search benchmarks are derived from the Einstein@Home multi-directed search for CWs in O1 data (cf. [253]). They cover a frequency range of 20-1500 Hz and large ranges of first and second-order spindown, which are functions of the characteristic age of the targeted supernova remnant and the frequency (see Table 8.2). Compared to the original search, however, the total observation time is substantially reduced to the two benchmark spans of $T = 10^5$ s and $T = 10^6$ s.

Similarly to [30], we limit the required matched-filtering computing cost by only searching a narrow frequency band of $\Delta f = 50$ mHz at a few representative starting frequencies in the range of 20-1500 Hz. The characteristics of the matched-filtering searches can be found in Table 8.3.

parameter name	parameter values
data span	$T = 10^5$ s / $T = 10^6$ s
detectors	LIGO Hanford (H1) + Livingston (L1)
noise	stationary white Gaussian ¹
sky-region	all-sky
frequency band	$f \in [20, 1000]$ Hz
spindown range	$\dot{f} \in [-10^{-10}, 0]$ Hz/s

Table 8.1.: Definition of all-sky (two-detector) benchmark searches.

parameter name	parameter values
data span	$T = 10^5$ s / $T = 10^6$ s
detectors	LIGO Hanford (H1) + Livingston (L1)
noise	stationary white Gaussian ¹
sky-position	CasA / G347
frequency band	$f \in [20, 1500]$ Hz
spindown range	$-f/\tau \leq \dot{f} \leq 0$ Hz/s
second order spindown	$0 \text{ Hz/s}^2 \leq \ddot{f} \leq 5f/\tau^2$
characteristic age τ	CasA: 330 yrs, G347: 1600 yrs

Table 8.2.: Definition of directed benchmark searches, modeled after [253].

search	mismatch	mean SNR loss	templates
all-sky $T = 10^5$ s	0.1	4%	$7 \cdot 10^{11}$
all-sky $T = 10^6$ s	0.2	8%	$4 \cdot 10^{14}$
G347 $T = 10^5$ s	0.1	5%	$1 \cdot 10^{10}$
G347 $T = 10^6$ s	0.2	10%	$6 \cdot 10^{12}$
CasA $T = 10^5$ s	0.1	5%	$6 \cdot 10^{10}$
CasA $T = 10^6$ s	0.2	10%	$3 \cdot 10^{13}$

Table 8.3.: WEAVE coherent matched-filtering search parameters and characteristics.

¹Excluding the real-data tests in Sec. 8.5.2

$\mathcal{D}_{\text{MF}}^{90\%} [\text{Hz}^{-1/2}]$	20 Hz	100 Hz	200 Hz	500 Hz	1000 Hz
a-s $T = 10^5$ s	16.0	15.0	14.5	14.2	13.6
a-s $T = 10^6$ s	42.0	40.1	39.4	38.3	35.9
$\mathcal{D}_{\text{MF}}^{90\%} [\text{Hz}^{-1/2}]$	20 Hz	500 Hz	1000 Hz	1500 Hz	
G347 $T = 10^5$ s	18.5	17.1	–	16.5	
G347 $T = 10^6$ s	46.1	43.9	–	43.1	
CasA $T = 10^5$ s	17.5	16.3	–	15.7	
CasA $T = 10^6$ s	46.1	43.8	43.4	–	

Table 8.4.: Sensitivity depths $\mathcal{D}_{\text{MF}}^{90\%}$ achieved by the WEAVE coherent matched-filtering search for the (two-detector) all-sky (a-s) and directed cases defined in Tables. 8.1 and 8.2. The all-sky sensitivity is improved by a factor of approximately $\sqrt{2}$ compared to the single-detector values reported in [30], as expected for coherent matched filtering. As training the CasA case $T = 10^6$ s, $f_0 = 1500$ Hz, “H1L1” required more GPU memory than available to us, we reduced the maximum frequency in the search to 1000 Hz.

8.3. Deep-Learning CWs

The approach used here is an evolved version of our previous deep-learning study in [30]: We train a noise-versus-signal classifier on strain data from two detectors. The input is provided as two separate channels per detector, each containing respectively the real and imaginary part of the Fourier transform of the strain data. This results in four input channels for our two-detector cases.²

However the networks could easily be trained for data from more detectors by adding additional channels. As for matched-filtering the additional data would increase the computational cost. Due to this straight forward generalization we only consider two-detectors as most matched-filtering searches at the moment only consider data from the two most sensitive detectors (e.g. see [28]).

We determine the maximal width in frequency of the signals in the allowed parameter space of the search and set the DNN input size to twice this width. This allows us to slide half overlapping windows over the frequency band to guarantee any signal is fully contained in at least one window. This leads to an increase of the DNN input size with observation time as well as with the number of detectors.

8.3.1. Finding a Network Architecture

We started experimenting with the modified 1D-ResNet architecture from [30] and other 1D-versions of architectures like *InceptionResNet-v2* [175] which have proven successful for image recognition. For various different architectures we trained a network on a smaller number of samples for the $T = 10^5$ s, $f_0 = 1000$ Hz, “H1L1” benchmark case. We compare the different networks’ performance by calculating their detection probability on the validation set described in Sec. 8.3.2.

The architecture with the best detection probability in these experiments is an Inception-ResNet architecture: The InceptionResNet-v2 architecture was modified to feature one-dimensional inputs. For the $T = 10^5$ s cases this network was further enlarged by increasing the number of block repetitions by 2, in width by increasing the filter sizes by 2 as well as the number of filters in every convolutional layer by a factor of 4. The resulting network needs

²Note that a neural network with one input dimension and multiple channels is still commonly referred to as a 1D network.

too much memory³ for the larger inputs of the $T = 10^6$ s cases, therefore we use the original non-enlarged network for the $T = 10^6$ s benchmark cases.

The DNN input is first normalized by its standard deviation. The DNN output is created with a global average pooling layer and a dense layer with two final neurons and a softmax activation. The two outputs are encoding the estimated probabilities that the input contains a signal in noise p_{signal} or pure noise, $p_{\text{noise}} = 1 - p_{\text{signal}}$, respectively.

The DNN was implemented in TENSORFLOW 2.0 [249] with its inbuilt Keras API (`tf.keras`). The CW signal generation was performed using the python SWIG-wrapping [255] of lalsuite [250].

8.3.2. DNN Training and Validation

We trained a total of 25 networks, one for each case listed in Table 8.4 and in addition one all-sky and two directed networks trained for the entire respective search frequency range of the $T = 10^5$ s second searches. As in [30] each of the networks is trained with a set of synthesized input vectors, where half contain pure Gaussian noise, and half contain a signal added to the noise. The training set is built from 5 000 pre-computed signals which are added to 24 dynamically generated noise realizations each. The noise realizations are also added as pure noise examples giving 240 000 samples in the training set in total.

The number of 5 000 signals was determined as a compromise between requirements in computing resources and the diminishing improvements which could be gained with a bigger training set (cf. [30] for details).

The signals are scaled to an evolving depth $\mathcal{D}_{\text{training}}(\text{epoch})$ which starts low, i.e. with louder signals, and then increases every five epochs until it reaches the final training depth $\mathcal{D}_{\text{training}}$, according to the following curriculum:

$$\mathcal{D}_{\text{training}}(\text{epoch}) = \frac{\mathcal{D}_{\text{training}}}{\gamma(\text{epoch mod } 5)}, \quad (8.2)$$

where $\gamma(n) = (1.75, 1.5, 1.25, 1, 1, 1, \dots)$, i.e. the signals get weaker until, after 15 epochs, the sensitivity depth reaches $\mathcal{D}_{\text{training}}$, which is chosen as the semi-analytic estimate for the WEAVE-sensitivity depth $\mathcal{D}_{\text{MF}}^{90\%}$, using the method of Ref. [28]. At the time of training the final measured WEAVE matched-filtering sensitivity depths of Table. 8.4 were not yet available, which is why we used the faster (but less accurate) sensitivity-estimation instead.

This type of curriculum learning [256] is necessary to teach the network to find the weak signals at the final depth. Without it the network seemed unable to pick up the weak signals at the beginning and therefore was unable to learn at all. This is probably a consequence of the vastly increased number of parameters in the network compared to the network used in [30].

The DNNs were trained with a categorical cross entropy loss and the Adadelta optimizer [164]. They were each trained for 10 days on NVIDIA GPUs (RTX2060,70,80(Ti), GV100, GTX 1660(Ti)) contained in the ATLAS computing cluster. By that time all the networks were fully trained, i.e. they did not show any significant improvement over the last couple of epochs.

During training we perform a validation step every five epochs where the detection probability is calculated on 20 000 independently-generated samples: 10 000 pure noise samples and 10 000 samples containing signals in Gaussian noise of the fixed depth $\mathcal{D}_{\text{training}}$.

In order to avoid a numerical overflow in the final softmax activation layer, we do not use the estimated softmax probabilities as a detection statistic. Instead we directly use the final linear network output which corresponds to p_{signal} (i.e. the respective input to the final softmax activation) as a detection statistic. The detection probability is calculated in the usual way as the fraction of signal cases where this statistic crosses over the $p_{\text{fa}} = 1\%$ threshold.

³The largest GPU used (NVIDIA Quadro GV100) has 32GB of GPU memory.

8.4. Characterizing DNN Performance on Gaussian Noise

As the networks’ parameters have been optimized for the training set and the network architecture (or *hyperparameters*) was optimized for the validation set, we need to evaluate the network’s performance on an independent *test set* to fully characterize its performance as a CW detection method. This test set consists of noise and signals with randomly drawn parameters from a distribution isotropic in the sky and uniform in the other parameters. It is generated on-the-fly using the LALSUITE software library [250, 255].

8.4.1. Detection Probabilities at Fixed False Alarm

The results in the following are presented in two ways:

1. the detection probability $p_{\text{det}}^{\text{DNN}}$ obtained at false alarm $p_{\text{fa}} = 1\%$ per 50 mHz frequency band for a signal population of fixed depth $\mathcal{D}_{\text{MF}}^{90\%}$, for which the coherent WEAVE matched-filtering search achieves $p_{\text{det}} = 90\%$.
2. The “upper-limit” depth $\mathcal{D}_{\text{DNN}}^{90\%}$ for the network, where it achieves a detection probability of $p_{\text{det}}^{\text{DNN}} = 90\%$ at $p_{\text{fa}} = 1\%$ per 50 mHz frequency band.

The measured DNN sensitivity on the all-sky search benchmarks is given in Tables 8.5 and 8.6. Similar to the previous single-detector results in [30], for $T = 10^5$ s at low frequencies the DNN achieves a performance close to matched filtering, while it increasingly falls behind for higher frequencies. However, for the $T = 10^6$ s cases the new network does not perform well and quickly drops to low sensitivity at increasing frequency.

The measured DNN sensitivity for the directed search benchmarks is also given in Tables 8.5 and 8.6. The results are similar in nature to the all-sky search results. For the $T = 10^5$ s searches for both targets the DNN gets relatively close to the matched-filtering performance, while for the $T = 10^6$ s searches they rapidly lose sensitivity when going to higher frequencies.

Note that in the $T = 10^6$ s searches our new network seems to perform worse and fall off more rapidly compared to the previous benchmark results in [30]. This loss in performance at $T = 10^6$ s can be traced back to two reasons: First, the new network architecture was optimized only for the $T = 10^5$ s searches, and second we only trained a single network instance instead of picking the best from an ensemble of 100 networks, as was done in [30], due to the increased hardware requirements of the new network architecture.

$\mathcal{D}_{\text{DNN}}^{90\%} [\text{Hz}^{-1/2}]$	20 Hz	100 Hz	200 Hz	500 Hz	1000 Hz
a-s $T = 10^5$ s	14.9	13.2	12.4	10.6	49.0
a-s $T = 10^6$ s	29.6	17.5	13.9	9.7	7.9
$\mathcal{D}_{\text{DNN}}^{90\%} [\text{Hz}^{-1/2}]$	20 Hz	500 Hz	1000 Hz	1500 Hz	
G347 $T = 10^5$ s	16.3	13.6	–	11.1	
G347 $T = 10^6$ s	33.9	11.7	–	1.3	
CasA $T = 10^5$ s	16.4	13.4	–	11.5	
CasA $T = 10^6$ s	28.1	0.0 ⁵	1.4	–	

Table 8.5.: Network sensitivity depths $\mathcal{D}_{\text{DNN}}^{90\%}$ for the (two-detector) all-sky (a-s) and directed search cases. The corresponding matched-filtering sensitivity depths are given in Table 8.4. As training the CasA case $T = 10^6$ s, $f_0 = 1500$ Hz, “H1L1” required more GPU memory than available to us, we reduced the maximum frequency in the search to 1000 Hz.

$p_{\text{det}}^{\text{DNN}} [\%]$	20 Hz	100 Hz	200 Hz	500 Hz	1000 Hz
a-s $T = 10^5$ s	$84.4^{+4.0}_{-2.3}$	$79.5^{+3.3}_{-3.5}$	$78.1^{+3.3}_{-2.9}$	$70.4^{+3.3}_{-3.4}$	$659.1^{+4.4}_{-3.7}$
a-s $T = 10^6$ s	$60.5^{+3.7}_{-3.1}$	$24.5^{+3.1}_{-3.1}$	$11.2^{+3.1}_{-2.4}$	$3.3^{+2.4}_{-1.3}$	$0.7^{+0.7}_{-0.8}$
$p_{\text{det}}^{\text{DNN}} [\%]$	20 Hz	500 Hz	1000 Hz	1500 Hz	
G347 $T = 10^5$ s	$79.6^{+3.1}_{-3.1}$	$71.8^{+5.1}_{-7.7}$	–	$64.2^{+3.6}_{-3.6}$	
G347 $T = 10^6$ s	$71.2^{+3.1}_{-3.0}$	$2.6^{+2.1}_{-1.2}$	–	$0.4^{+1.1}_{-0.6}$	
CasA $T = 10^5$ s	$86.4^{+3.3}_{-5.5}$	$75.2^{+3.1}_{-4.4}$	–	$65.5^{+3.4}_{-3.6}$	
CasA $T = 10^6$ s	$54.6^{+3.3}_{-3.7}$	$0.6^{+0.6}_{-0.7}$	$0.7^{+1.0}_{-0.7}$	–	

Table 8.6.: Network detection probabilities $p_{\text{det}}^{\text{DNN}}$ with 95 % error region for the (two-detector) all-sky (a-s) cases and directed cases for signals at the matched-filtering sensitivity depths $\mathcal{D}_{\text{MF}}^{90\%}$ given in Table 8.4. As training the CasA case $T = 10^6$ s, $f_0 = 1500$ Hz, “H1L1” required more GPU memory than available to us, we reduced the maximum frequency in the search to 1000 Hz.

8.4.2. Generalization

One of most promising features of the DNN benchmarks results found in [30] was the surprising capability of the DNN to generalize to signal parameters it was not trained for. We confirm this feature for the new DNN used for the $T = 10^5$ s all-sky searches in this work for frequency, signal strength, spindowns and sky position. For the $T = 10^5$ s directed search benchmarks introduced in this work, we also find a remarkable capability to generalize despite being less general than the all-sky DNNs. Given the rather poor DNN performance on the $T = 10^6$ s cases, discussed in Sec. 8.4.1, we do not include those in the generalization tests shown here.

Frequency

To avoid large computational cost for the training, we want to use as few networks as possible, optimally even a single one, to cover the search band with a reasonable sensitivity. Therefore we want to compare how a network trained over the full frequency band compares to “specialized” narrow-band networks trained on 50 mHz bands, when tested over the full frequency range.

The results of these tests for the all-sky two detector $T = 10^5$ s search can be found in Fig. 8.1. We find that the “specialized” networks trained for small frequency bands generalize well to lower frequency and slightly worse but still quite well to higher frequencies, confirming the findings in the single-detector case in [30]. However, the network trained over the full frequency range shows promise as it seems to fall only marginally behind the specialized networks for most frequencies – even beating some of the specialized networks at their training frequencies.

In the case of directed-search DNNs shown in Fig. 8.2, we see much narrower generalization around the trained frequencies of the “specialized” networks compared to the all-sky cases. The better generalization of the all-sky networks is likely due to the (known) near-degeneracy between frequency and sky position for short observation times. The networks trained over the full frequency in the directed cases significantly fall behind the specialized networks at their respective frequencies, contrary to our finding in the all-sky case. This is also likely connected

⁴The given result is from a network trained on the whole frequency range, the specialized network performed worse, having a sensitivity depth of $7.9 \text{ Hz}^{-1/2}$ (see Fig. 7.6).

⁵The network did not reach 90 % detection probability even at the lowest depth tested $\mathcal{D}^{90\%} = 0.1 \text{ Hz}^{-1/2}$

⁶The given result is from a network trained on the whole frequency range, the specialized network performed worse, reaching a detection probability of $47.9^{+4.0}_{-3.8}\%$ (see Fig. 7.6).

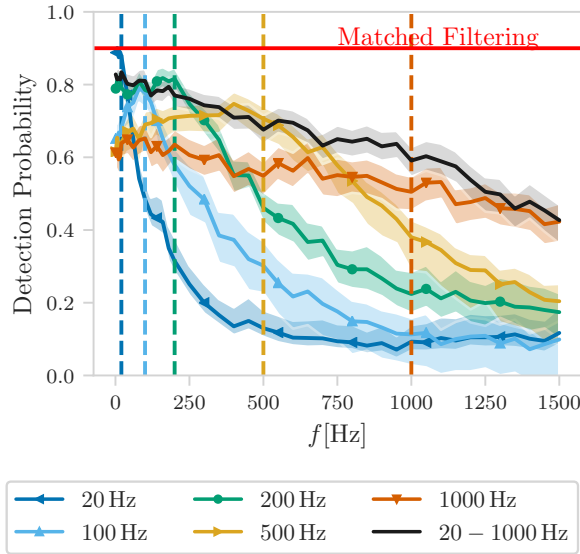


Figure 8.1.: Detection probability p_{det} versus injection frequency f for the all-sky networks trained at five different frequencies and for a network trained with signals drawn from the full frequency range (solid black line). The dashed vertical lines mark the respective training frequencies for the five “specialized” networks. The solid red horizontal line represents the coherent matched-filtering detection performance. The shaded areas around each curve show the 95% error regions. The analogous single-detector result is found in Fig.6a of [30].

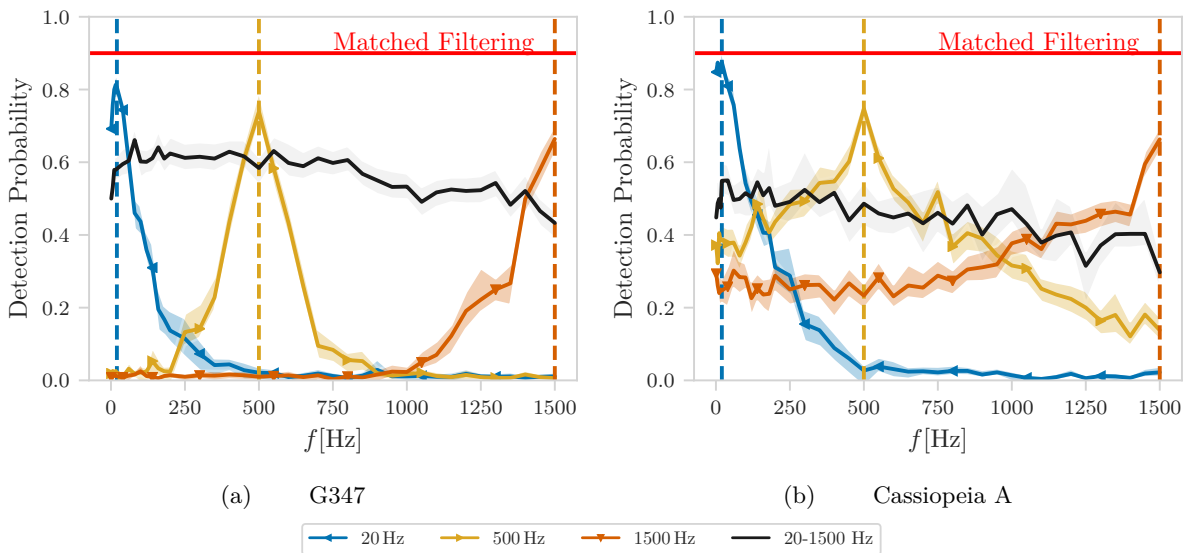


Figure 8.2.: Detection probability p_{det} versus injection frequency f for networks trained at three different frequencies for the CasA and the G347 target, respectively, and for a network trained with signals drawn from the full frequency range (solid black line). The dashed vertical lines mark the respective training frequencies for the three networks. The solid red horizontal line represents the coherent matched-filtering detection performance. The shaded areas around each curve show the 95% error regions.

to the mentioned near-degeneracy, as in the directed case, increasing the frequency range forces the network to learn many new signal shapes, while in the all-sky case the new signal shapes were already covered via signals from different sky-positions.

Signal Strength

To fully characterize a search method it is important to look at the detection efficiency curve, i.e. the detection probability for different signal strengths, shown in Fig. 8.3. This is especially interesting given that we use a single (final) depth $\mathcal{D}_{\text{training}}$ for training (cf. Sec. 8.3.2). The observed efficiency curves are very similar across the different searches, hence we only show two representative examples, the directed CasA search at $T = 10^5$ s, $f_0 = 20$ Hz, “H1L1”, and an all-sky search at $T = 10^5$ s, $f_0 = 1000$ Hz, “H1L1”.

In general the DNNs show qualitatively similar efficiency curves as matched-filter searches. We notice especially that the curves become almost indistinguishable for the low frequency cases while for higher frequency the DNNs relations seems to be shifted towards their overall lower sensitivity.

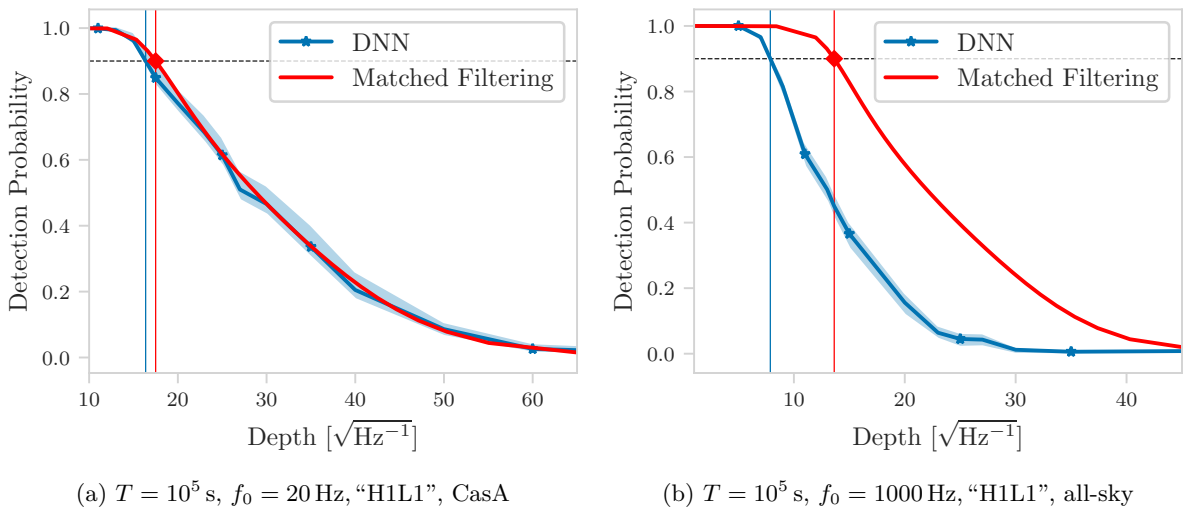


Figure 8.3.: Detection probability p_{det} versus injection depth \mathcal{D} for networks trained on the respective matched-filtering depth $\mathcal{D}_{\text{MF}}^{90\%}$ (indicated by the vertical solid line with the diamond at 90%). The second vertical line which crosses the DNN curve at 90% gives the sensitivity depth for the DNN at 90% detection probability. The shaded region around the DNN curve is the 95% error region. The respective errors for the matched-filtering results are smaller than the thickness of the curve.

Spindowns

Another important aspect we want to consider is the detectability of signals with first and second order spindowns outside of the training range, shown in Figs. 8.4 and 8.5, respectively.

For the all-sky searches we observe a similar behavior (not shown) in \dot{f} to the results reported in Fig. 7 of [30] for the single-detector benchmarks: a plateau of nearly constant detection probability by far exceeding the training region. For the directed searches, however, we see a different behavior in \dot{f} , shown in Fig. 8.4: the DNNs plateau of nearly-constant detection probability falls off starting from the maximum absolute spindown value of the training set. The generalization is not completely symmetric, though, and extends to larger negative than positive spindowns. This might be an effect of the purely positive second-order spindown breaking the degeneracy. The strong generalization beyond trained spindowns of the all-sky DNNs might be

due to the (known) near-degeneracy between spindown and sky position for short observation times compared to a year.

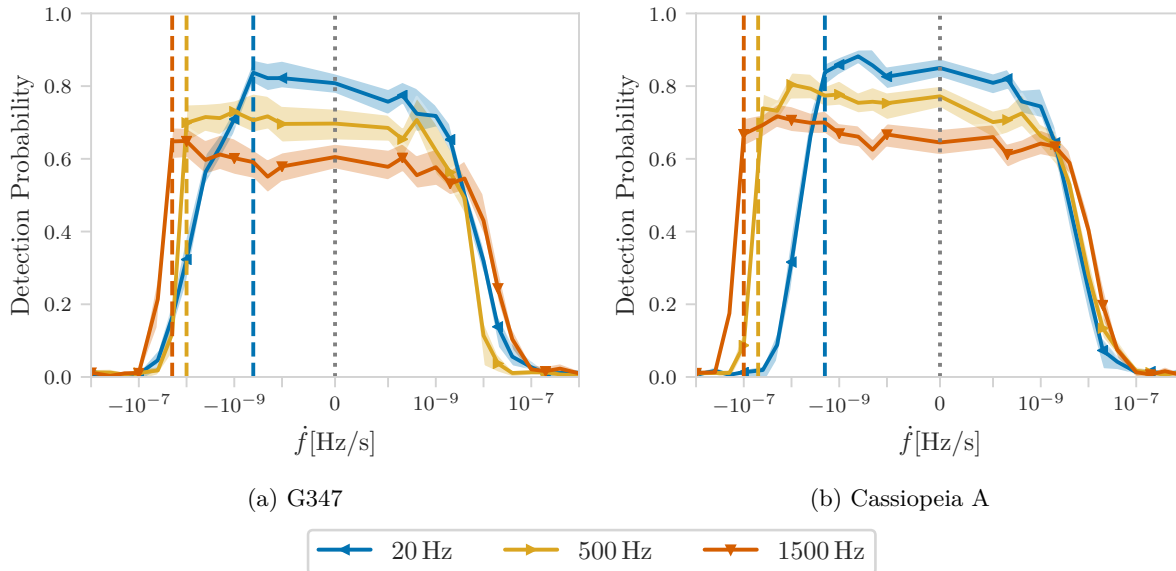


Figure 8.4.: Detection probability p_{det} versus injected spindown \dot{f} for networks trained at different frequencies. The x-axis is plotted as a symmetric logarithm, i.e. logarithmic for the larger negative values, linear for $|\dot{f}| < -10^{-10}$ Hz/s and logarithmic for the larger positive values. The vertical dashed lines mark the minimal spindown \dot{f} used in the training set. Its absolute value increases with frequency. The maximal used spindown for all cases is 0 Hz/s (dotted line). The shaded areas around each curve show the 95% error regions.

The generalization results on the second-order spindown of the directed searches in Fig. 8.5 show a qualitatively similar behavior to the first-order spindown: a plateau of nearly-constant detection probability and a drop starting at about $|\ddot{f}| \gtrsim 10^{-14}$ Hz/s², approaching zero near $|\ddot{f}| \gtrsim 10^{-11}$ Hz/s². Contrary to the first-order spindown results, however, the drop happens many orders of magnitude beyond the trained range of $\ddot{f} \lesssim 10^{-17}$ Hz/s². This is not surprising, given that a second-order spindown of this order would only change the signal phase by about 10^{-2} rad over the short timespan of $T = 10^5$ s and is therefore still effectively negligible.

Sky Position

Another interesting question is the sensitivity as a function of the sky-position of the signal. For this we measure and plot the DNN detection probability as a function of the sky-position of the signal injections, shown in Fig. 8.6.

Here we use signals injected at fixed SNR ($\rho = 8.94$) instead of the fixed-depth \mathcal{D} injections used in other tests. By fixing the signal SNR, we can probe the intrinsic sky-position sensitivity of the trained network independently of the detector antenna-patterns while for signals at fixed depth the SNR varies with sky position.

For the directed searches in Fig. 8.6a- 8.6d we see a clear preference for the trained sky-position, while sensitivity localization improves with frequency. This is qualitatively similar to how matched filtering behaves, but with a wider sensitive sky region around the targeted sky-position. For matched filtering we estimate the sensitive region to be of order ~ 1 rad at $f = 20$ Hz and $\sim 10^{-2}$ rad at $f = 1500$ Hz.

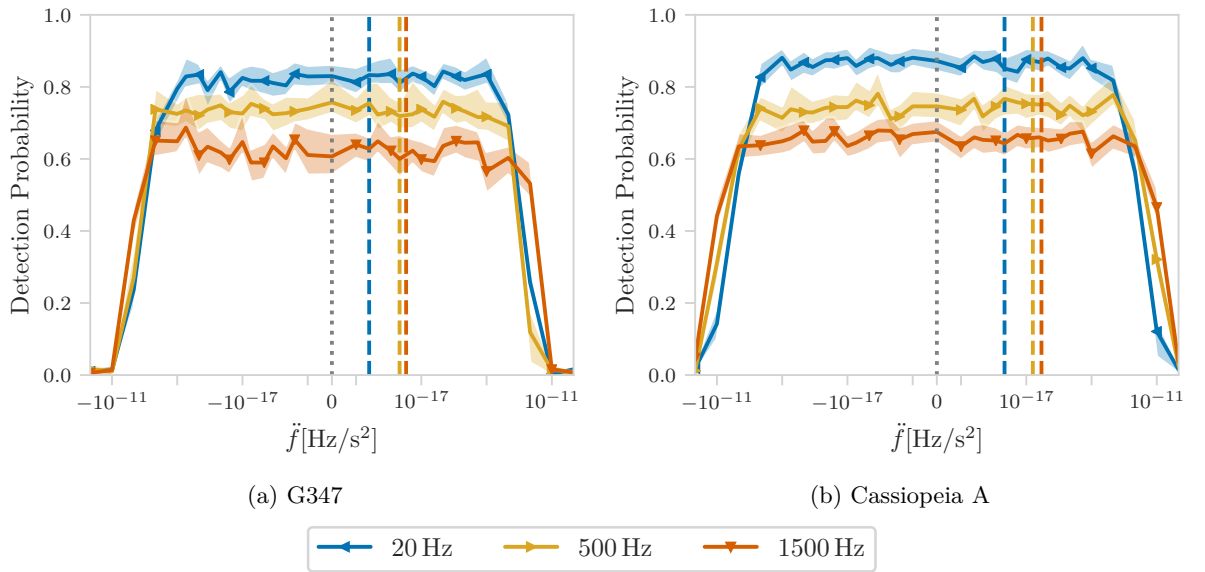


Figure 8.5.: Detection probability p_{det} versus injected second order spindown \ddot{f} for networks trained at different frequencies. All other parameters were the same as during training. The x-axis is plotted as a symmetric logarithm, i.e. logarithmic for the larger negative values, linear for $|\ddot{f}| < -10^{-20}$ Hz/s and logarithmic for the larger positive values. The vertical dashed lines mark the maximal second order spindown \ddot{f} used in the training set, which increases with frequency. The minimal used second order spindown for all cases is 0 Hz/s^2 (dotted line). The shaded areas around each curve show the 95% error regions.

8. Deep-Learning Continuous Gravitational Waves: Multiple detectors and realistic noise

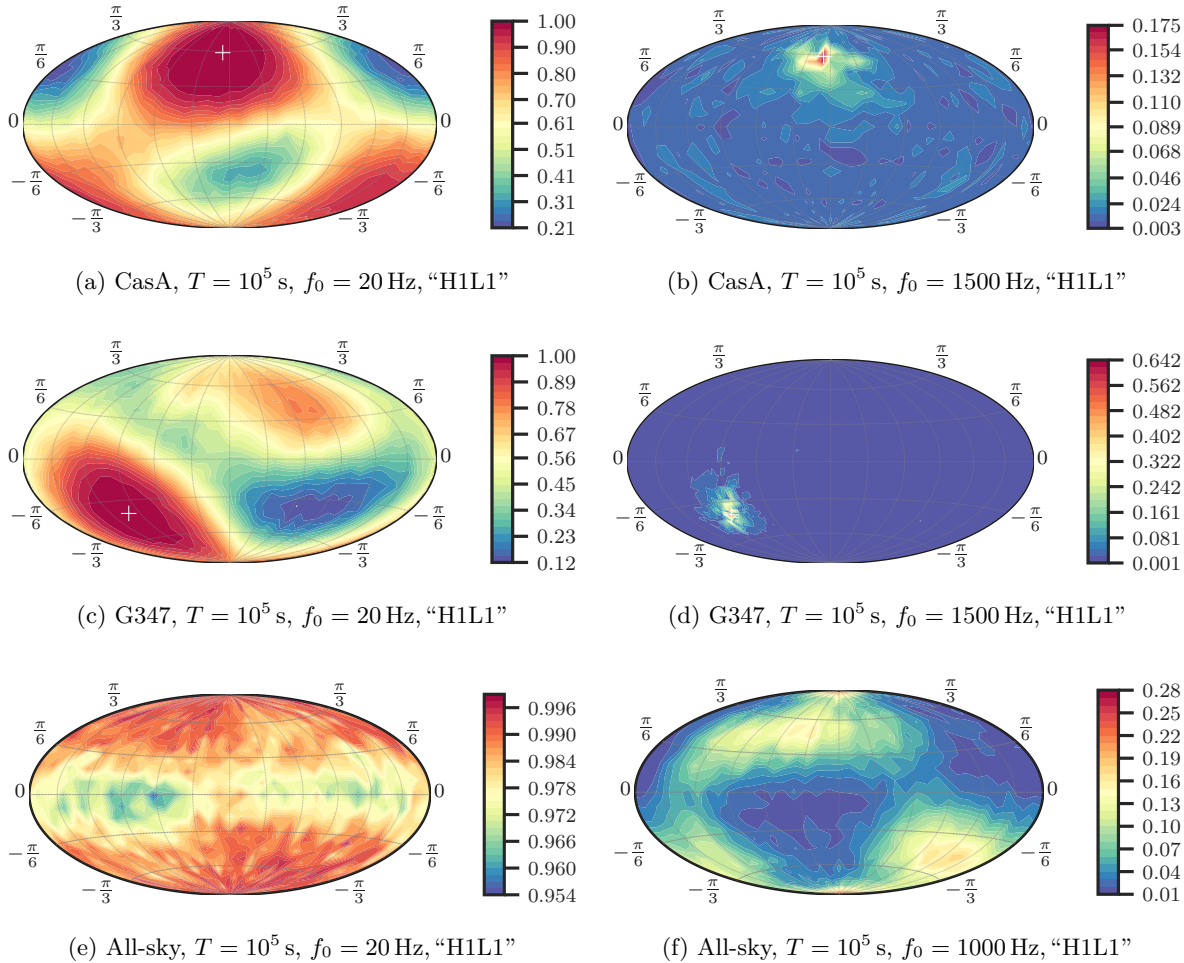


Figure 8.6.: Detection probability p_{det} as a function of the sky-position of injected signals in equatorial coordinates (Hammer projection). The detection probability is measured at fixed SNR $\rho = 8.94$. In (a)-(d) the respective sky position of CasA or G347 is marked by a white plus.

For the all-sky DNNs we see a preference for signals coming from the equatorial poles (latitude $\pm\pi/2$) instead, shown in Figs. 8.6e- 8.6f. In the $f = 20$ Hz case this effect is relatively small (with a difference of only $\sim 5\%$ in detection probability), and much more pronounced at $f = 1000$ Hz, where we see some additional structure in right ascension.

We suspect that the observed preference for signals coming from the poles is likely due to their smaller Doppler-broadening compared to signals from the equator, which makes them more concentrated in the frequency domain and therefore easier to “see” for the network. This is also consistent with the DNN detection probability decreasing with increasing signal frequency and increasing observation time, which both result in signals getting more spread out in frequency due to the increase in Doppler broadening.

8.5. Testing Network Performance on Real Data

In order to conduct a search for CWs with a DNN, the network has to be able to handle real detector data, which differs in three aspects from the simulated Gaussian data used so far in this study:

1. potentially different noise levels between detectors

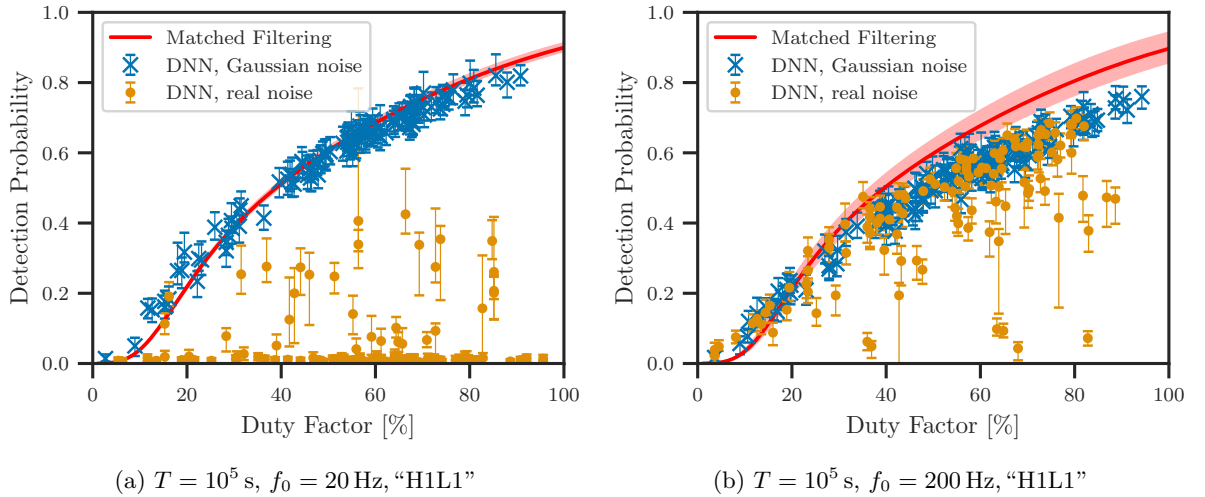


Figure 8.7.: Duty factor vs detection probability of an all-sky DNN in Gaussian and real noise. The solid red curve with its shaded region represents the behaviour of matched filtering on Gaussian noise, the blue crosses represent the DNN’s performance on Gaussian noise and the yellow circles represent the DNN’s performance on real LIGO O1 detector noise. The error bars indicate the 95 % confidence interval.

2. typically non-contiguous data, i.e. gaps in the data due to real gravitational-wave detectors not being in lock continuously
3. non-Gaussian disturbances in the data, in particular near-monochromatic *lines* that can mimic CWs and trigger false alarms (e.g. see [142] for more discussion).

Here we assess the impact of these effects on the detection performance of a DNN *trained on ideal simulated Gaussian noise without gaps*. In order to separate the different effects, we first test the DNN on simulated Gaussian noise with realistic data gaps, and then with real detector noise, both from a “quiet” undisturbed band and from a disturbed band. The next natural step would be to train networks directly on real detector noise, however this is beyond the scope of this work.

The detector data used is from the LIGO O1 observing run, which can be retrieved from the Gravitational Wave Open Science Center (GWOSC) [53].

8.5.1. Gaussian Noise with Data Gaps

In order to generate data with realistic gaps we randomly select start-times from the LIGO O1 run and retrieve the corresponding gap profile over $T = 10^5$ s. We then generate Gaussian white noise and signals with the same gaps, and we calculate the *duty factor* of this gap profile as $\frac{T_{\text{data}}}{2T} \leq 1$, where T_{data} is the amount of data from both detectors.

In Fig. 8.7 we show the results of detection probability as a function of duty factor for two test cases, namely the all-sky benchmarks for $T = 10^5$ s, $f_0 = 20$ Hz, “H1L1” and $T = 10^5$ s, $f_0 = 200$ Hz, “H1L1”. In both cases we see that the DNN’s detection probability (*cross markers*) shows a similar drop in detection probability with decreasing duty factor as matched-filtering does (*solid line*). This indicates that the loss in detection probability stems purely from the intrinsic lowered signal SNR (due to the reduced amount of data), despite the network being trained on fully-contiguous data only.

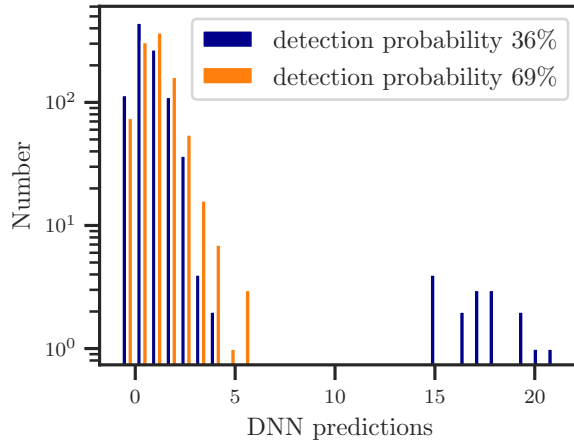


Figure 8.8.: Histogram of the distribution of DNN detection statistic values (predictions) for 1000 real-noise input samples. The two distributions correspond to two different start-times, with similar duty factors $\sim 82\%$ and for the same 5 Hz band around 200 Hz. In one case (blue) a disturbance in the data results in a long tail of higher statistic values, which leads to a higher detection threshold at fixed false-alarm, thereby reducing detection probability compared to the undisturbed case (orange).

8.5.2. Performance on Real Detector Data

For this test we use real strain data from the LIGO O1 observing run, with gating and cleaning applied for a recent Einstein@Home⁷ search [253].

For a time-span of $T = 10^5$ s with randomly-selected start-time during O1, we draw 1000 random 50 mHz-frequency bands from within a 5 Hz band around the test frequency. Using these data samples we determine the detection probability in the usual way: apply the DNN to the data samples (assumed to be pure noise) to determine the detection threshold at $p_{fa} = 1\%$, then repeat the procedure with added signals of depth $\mathcal{D}_{MF}^{90\%}$ in order to determine the detection probability (i.e. the fraction of signal samples where the DNN prediction exceeds the threshold).

We found that performing an additional pre-normalization of the data by the individual detector noise floors improves the DNN detection performance in the presence of differing noise floors between the two detectors.

The results of the real-data tests are shown in Fig. 8.7 (*filled circles*), plotted again as a function of duty factor. For the frequency band at $f = 200$ Hz in Fig. 8.7b there are many data points basically matching the Gaussian-noise performance, while for others there is a substantial loss in detection probability. This loss can be traced to the presence of “line” disturbances in the data as the disturbed bands create a longer-tailed distribution of DNN detection statistic values for noise inputs. This is illustrated in Fig. 8.8 for one example. In the low-frequency $f = 20$ Hz case in Fig. 8.7a we see a significant overall drop in detection probability, due to a large number of lines and other disturbances we observed in this frequency band.

8.6. Discussion

In this work we demonstrated that the already-established ability of a deep neural network to search for continuous gravitational waves in the data of a single detector can be extended to two-detector searches. While the larger size of the input data increases the challenge for the DNN the results for short data spans remain reasonably competitive with matched filtering.

⁷<https://einsteinathome.org>

On the other hand our architecture searches did not yet yield a reasonably competitive neural network for the longer $T = 10^6$ s data span. Therefore we mostly focused on characterizing the performance of the $T = 10^5$ s networks for now.

Also note that compared to state-of-the-art CW searches the DNN sensitivity achieved here is not yet competitive. For example all-sky searches roughly achieve a sensitivity depth of $30 - 50 \text{ Hz}^{-1/2}$ (e.g. see [28]) while directed searches go up to $54 - 83 \text{ Hz}^{-1/2}$ [253].

As was shown in [30] the computing cost of a neural network search is dominated by the training time and the time of a matched-filtering follow-up⁸. This implies that multiple reuses of a trained DNN do not significantly increase the overall computing cost. Training, executing and following-up the $T = 10^5$ s search, using the networks presented in this chapter, is roughly two times faster than the respective matched-filtering search.

Furthermore we studied the features of a DNN search directed at a specific sky-position. These directed searches show comparable performance to the all-sky searches at $T = 10^5$ s with respect to the respective matched filter sensitivities, but show less generalization in frequency and first-order spindown.

A common trend observed here, consistent with the previous study [30], is that the network performance seems to degrade when signals are spread over a wider frequency band, i.e. for higher frequencies, sky positions with more Doppler spreading, and for longer timespans. This shows that the networks still have difficulties learning this aspect of input signals.

We have further shown that DNNs seem relatively robust towards data gaps that differ from the training set, and we found that the impact of unequal detector noise floors can be alleviated by per-detector data normalization. Furthermore, as expected, we find that the performance on real detector noise is significantly reduced in the presence of non-Gaussian disturbances, i.e. “lines”.

We can identify the following remaining steps towards a competitive and practical DNN search method:

1. Train the networks on real detector data in order to “learn” to classify disturbances as noise.
2. Further optimize network architecture to further close the gap to matched filtering under data ideal conditions.
3. Design a “semi-coherent”-type search method by combining the DNN predictions from short time spans (such as $T = 10^5$ s).

⁸The matched-filtering follow-up is currently necessary for a fair comparison as otherwise a matched-filtering search would deliver far more information about candidates than the DNN.

9. Outlook

With the first detection of gravitational waves from binary-black-hole and binary-neutron-star mergers, gravitational-wave research entered the new era of gravitational-wave astronomy. A detection of continuous gravitational waves (CWs) could contribute to gravitational-wave astronomy with insights about neutron stars and maybe even more fundamental principles, like the existence of axions (as we saw in Sec. 3.4.2).

Searches for continuous gravitational waves are getting more and more sensitive but even the current most sensitive searches for known pulsars do not yet reach the estimated necessary sensitivity as illustrated in Fig. 3.7. Therefore, wide-parameter searches for unknown pulsars might be the most promising search type as they could detect the one neutron star radiating strongly enough to be detected with the current advanced LIGO detectors.

However, wide-parameter CW searches are severely limited by computational cost. The high computational cost often makes it necessary to efficiently estimate the sensitivity of searches to sensibly determine search parameters before launching the search and to verify the upper limits. In the course of this work, a faster and more accurate sensitivity estimator was developed (see Ch. 6). This estimator was also used when we explored a new approach to CW searches: deep neural networks (see Chs. 7 and 8). We saw that deep neural networks (DNNs) can detect CWs for small amounts of data under the idealized circumstances of white Gaussian noise. We also saw that DNNs need further research in order to become a competitive search method. During the work on this thesis we also tested some alternative approaches and improvements.

For example, we trained networks to *denoise* gravitational-wave data. In this case the labels for the training set are not whether the data contains a signal or not but just the pure injected signals, i.e. for noise all labels are zero. While this approach did show some promise and was also explored for signals from merging binaries in [257], it was not clear how to use the denoised data efficiently and the straight-forward training for a detection statement seemed more promising.

Furthermore, an exploration of “semi-coherent” deep-learning searches for CWs was started. In this approach, a longer timespan of data, e.g. $T = 10^6$ s, is divided into segments of a shorter timespan, e.g. $T = 10^5$ s. A network trained on the short timespan, as presented in Chs. 7 and 8, is then applied to every segment. In each segment the network has also to be slid over the frequency range, giving a time-frequency map of network predictions as output.

These outputs can now be combined to form a new detection statistic on data of the longer timespan. This can be done e.g. by summing as is used in the classical StackSlide method to combine \mathcal{F} -statistic values (presented in Sec. 4.4.5). However, in our first tests a secondary neural network seemed to be able to reach better performance. This network is trained on the time-frequency map outputs of the first network to predict if there is a signal in the data or not. In our first tests we used a dense network with 3 hidden layers of 32 neurons each and single-detector data. This small network was able to give a better performance than the “coherent” network for the $T = 10^6$ s, $f_0 = 1000$ Hz benchmark from Ch. 7. The “semi-coherent” pipeline improved the sensitivity depth of $\mathcal{D}^{90\%} = 8.9 \text{ Hz}^{-1/2}$ to $\mathcal{D}^{90\%} \gtrsim 12 \text{ Hz}^{-1/2}$. However, this is still far less sensitive than the coherent matched-filtering result at $26 \text{ Hz}^{-1/2}$.

Further research will reveal if the DNN performance can be improved enough so that their incredible speed advantage can eventually create more sensitive searches than matched-filtering. Simultaneously matched-filtering methods are also improved upon continuously. It therefore remains to be seen which method will ultimately lead to the first detection of a continuous wave, catapulting the field of CW data analysis into the exciting future of gravitational-wave astronomy.

A. Bibliography

- [1] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116.6, 061102 (Feb. 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. arXiv: 1602.03837 [gr-qc].
- [2] B. P. Abbott et al. “GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral”. In: *Phys. Rev. Lett.* 119.16, 161101 (Oct. 2017), p. 161101. DOI: 10.1103/PhysRevLett.119.161101. arXiv: 1710.05832 [gr-qc].
- [3] B. P. Abbott et al. “Multi-messenger Observations of a Binary Neutron Star Merger”. In: *The Astrophysical Journal* 848.2 (Oct. 2017), p. L12. DOI: 10.3847/2041-8213/aa91c9. URL: <https://doi.org/10.3847/2F2041-8213/2Faa91c9>.
- [4] B. P. Abbott et al. “A gravitational-wave standard siren measurement of the Hubble constant”. In: *Nature* 551 (Nov. 2017), pp. 85–88. DOI: 10.1038/nature24471. arXiv: 1710.05835. URL: <https://www.nature.com/articles/nature24471>.
- [5] MR Drout et al. “Light curves of the neutron star merger GW170817/SSS17a: Implications for r-process nucleosynthesis”. In: *Science* 358.6370 (2017), pp. 1570–1574. URL: <https://science.sciencemag.org/content/358/6370/1570>.
- [6] Darach Watson et al. “Identification of strontium in the merger of two neutron stars”. In: *Nature* 574.7779 (2019), pp. 497–500. URL: <https://www.nature.com/articles/s41586-019-1676-3>.
- [7] *Einstein@Home Project Page*. URL: <https://einsteinathome.org/>.
- [8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521 (2015), pp. 436–. URL: <http://dx.doi.org/10.1038/nature14539>.
- [9] D. George and E. A. Huerta. “Deep Neural Networks to Enable Real-time Multimessenger Astrophysics”. In: *Phys. Rev. D.* 97 (2018), p. 044039. DOI: 10.1103/PhysRevD.97.044039. arXiv: 1701.00008v3 [astro-ph.IM].
- [10] H. Gabbard et al. “Matching matched filtering with deep networks in gravitational-wave astronomy”. In: *Phys. Rev. Lett.* 120 (2018), p. 141103. arXiv: 1712.06041 [astro-ph.IM].
- [11] D. George and E. A. Huerta. “Deep Learning for Real-time Gravitational Wave Detection and Parameter Estimation: Results with Advanced LIGO Data”. In: *Physics Letters B* 778 (2018), pp. 64–70. DOI: 10.1016/j.physletb.2017.12.053. arXiv: 1711.03121 [gr-qc].
- [12] T. Gebhard et al. “ConvWave: Searching for Gravitational Waves with Fully Convolutional Neural Nets”. In: *Workshop on Deep Learning for Physical Sciences (DLPS) at the 31st Conference on Neural Information Processing Systems (NIPS)*. 2017. URL: https://dl4physicalsciences.github.io/files/nips_dlps_2017_13.pdf.
- [13] X. Fan et al. “Applying deep neural networks to the detection and space parameter estimation of compact binary coalescence with a network of gravitational wave detectors”. In: *Science China Physics, Mechanics, and Astronomy* 62, 969512 (June 2019), p. 969512. DOI: 10.1007/s11433-018-9321-7. arXiv: 1811.01380 [astro-ph.IM].

A. Bibliography

- [14] T. D. Gebhard et al. “Convolutional neural networks: A magic bullet for gravitational-wave detection?” In: *Phys. Rev. D* 100 (6 Sept. 2019), p. 063015. DOI: 10.1103/PhysRevD.100.063015. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.063015>.
- [15] He Wang et al. “Gravitational wave signal recognition of O1 data by deep learning”. In: (2019). arXiv: 1909.13442 [astro-ph.IM].
- [16] Andrew L. Miller et al. “How effective is machine learning to detect long transient gravitational waves from neutron stars in a real search?” In: *Phys. Rev. D* 100 (6 Sept. 2019), p. 062005. DOI: 10.1103/PhysRevD.100.062005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.062005>.
- [17] Filip Morawski, Michał Stanisław Bejger, and Paweł Ciecieląg. “Convolutional neural network classifier for the output of the time-domain F-statistic all-sky search for continuous gravitational waves”. In: *Machine Learning: Science and Technology* (2020). URL: <https://iopscience.iop.org/article/10.1088/2632-2153/ab86c7>.
- [18] B. Beheshtipour and M. A. Papa. “Deep learning for clustering of continuous gravitational wave candidates”. In: *Phys. Rev. D* 101.6, 064009 (Mar. 2020), p. 064009. DOI: 10.1103/PhysRevD.101.064009. arXiv: 2001.03116 [gr-qc].
- [19] H. Gabbard et al. “Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy”. In: (2019). arXiv: 1909.06296 [astro-ph.IM].
- [20] Stephen R. Green, Christine Simpson, and Jonathan Gair. “Gravitational-wave parameter estimation with autoregressive neural network flows”. In: (2020). arXiv: 2002.07656 [astro-ph.IM].
- [21] N. Mukund et al. “Transient classification in LIGO data using difference boosting neural network”. In: *Phys. Rev. D* 95.10, 104059 (May 2017), p. 104059. DOI: 10.1103/PhysRevD.95.104059. arXiv: 1609.07259 [astro-ph.IM].
- [22] M. Zevin et al. “Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science”. In: *Classical and Quantum Gravity* 34.6, 064003 (Mar. 2017), p. 064003. DOI: 10.1088/1361-6382/aa5cea. arXiv: 1611.04596 [gr-qc].
- [23] M. Razzano and E. Cuoco. “Image-based deep learning for classification of noise transients in gravitational wave detectors”. In: *Classical and Quantum Gravity* 35.9, 095016 (May 2018), p. 095016. DOI: 10.1088/1361-6382/aab793. arXiv: 1803.09933 [gr-qc].
- [24] D. George, H. Shen, and E. A. Huerta. “Classification and unsupervised clustering of LIGO data with Deep Transfer Learning”. In: *Phys. Rev. D* 97.10, 101501 (May 2018), p. 101501. DOI: 10.1103/PhysRevD.97.101501. arXiv: 1706.07446 [gr-qc].
- [25] S. Vinciguerra et al. “Enhancing the significance of gravitational wave bursts through signal classification”. In: *Classical and Quantum Gravity* 34.9, 094003 (May 2017), p. 094003. DOI: 10.1088/1361-6382/aa6654. arXiv: 1702.03208 [astro-ph.IM].
- [26] P. Astone et al. “New method to observe gravitational waves emitted by core collapse supernovae”. In: *Phys. Rev. D*. 98.12, 122002 (Dec. 2018), p. 122002. DOI: 10.1103/PhysRevD.98.122002. arXiv: 1812.05363 [astro-ph.IM].
- [27] S. Klimentenko et al. “Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors”. In: *Phys. Rev. D*. 93.4, 042004 (Feb. 2016), p. 042004. DOI: 10.1103/PhysRevD.93.042004. arXiv: 1511.05999 [gr-qc].
- [28] Christoph Dreissigacker, Reinhard Prix, and Karl Wette. “Fast and accurate sensitivity estimation for continuous-gravitational-wave searches”. In: *Phys. Rev. D* 98 (8 Oct. 2018), p. 084058. DOI: 10.1103/PhysRevD.98.084058. URL: <https://link.aps.org/doi/10.1103/PhysRevD.98.084058>.

- [29] Karl Wette. “Estimating the sensitivity of wide-parameter-space searches for gravitational-wave pulsars”. In: *Phys. Rev. D*. 85.4 (2012), p. 042003. DOI: 10.1103/PhysRevD.85.042003.
- [30] Christoph Dreissigacker et al. “Deep-learning continuous gravitational waves”. In: *Phys. Rev. D* 100.4, 044009 (Aug. 2019), p. 044009. DOI: 10.1103/PhysRevD.100.044009. arXiv: 1904.13291 [gr-qc].
- [31] Christoph Dreissigacker and Reinhard Prix. “Deep-learning continuous gravitational waves: Multiple detectors and realistic noise”. In: *Phys. Rev. D* 102 (2 July 2020), p. 022005. DOI: 10.1103/PhysRevD.102.022005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.022005>.
- [32] James Clerk Maxwell. “VIII. A dynamical theory of the electromagnetic field”. In: *Philosophical transactions of the Royal Society of London* 155 (1865), pp. 459–512.
- [33] Oliver Heaviside. “A gravitational and electromagnetic analogy”. In: *The Electrician* 31.Part I (1893), pp. 281–282.
- [34] Henri Poincaré. *Sur la dynamique de l'électron*. Circolo Matematico di Palermo, 1906.
- [35] Albert Einstein. “The field equations of gravitation”. In: *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)* 1915 (1915), pp. 844–847.
- [36] Jorge Cervantes-Cota, Salvador Galindo-Uribarri, and George Smoot. “A Brief History of Gravitational Waves”. In: *Universe* 2.3 (Sept. 2016), p. 22. ISSN: 2218-1997. DOI: 10.3390/universe2030022. URL: <http://dx.doi.org/10.3390/universe2030022>.
- [37] Joel M Weisberg, Joseph H Taylor, and Lee A Fowler. “Gravitational waves from an orbiting pulsar”. In: *Scientific American* 245.4 (1981), pp. 74–83.
- [38] Charles W Misner, Kip S Thorne, John Archibald Wheeler, et al. *Gravitation*. Macmillan, 1973.
- [39] Bernard Schutz. *A first course in general relativity*. Cambridge university press, 2009.
- [40] Reinhard Prix and John T Whelan. “F-statistic search for white-dwarf binaries in the first Mock LISA Data Challenge”. In: *Classical and Quantum Gravity* 24.19 (2007), S565. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/24/19/S19>.
- [41] Kip S. Thorne. “Multipole expansions of gravitational radiation”. In: *Rev. Mod. Phys.* 52 (2 Apr. 1980), pp. 299–339. DOI: 10.1103/RevModPhys.52.299. URL: <https://link.aps.org/doi/10.1103/RevModPhys.52.299>.
- [42] Reinhard Prix. “Gravitational Waves from Spinning Neutron Stars”. In: *Neutron Stars and Pulsars*. Ed. by Werner Becker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 651–685. ISBN: 978-3-540-76965-1. DOI: 10.1007/978-3-540-76965-1_24. URL: https://doi.org/10.1007/978-3-540-76965-1_24.
- [43] B. P. Abbott et al. “LIGO: The Laser interferometer gravitational-wave observatory”. In: *Rept. Prog. Phys.* 72 (2009), p. 076901. DOI: 10.1088/0034-4885/72/7/076901. arXiv: 0711.3041 [gr-qc].
- [44] J. Aasi et al. “Advanced LIGO”. In: *Class. Quant. Grav.* 32, 074001 (Apr. 2015), p. 074001. DOI: 10.1088/0264-9381/32/7/074001.
- [45] T. Accadia et al. “Virgo: a laser interferometer to detect gravitational waves”. In: *Journal of Instrumentation* 7 (2012), P03012. DOI: 10.1088/1748-0221/7/03/P03012.
- [46] 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].

A. Bibliography

- [47] H. Grote. “The GEO 600 status”. In: *Class. Quant. Grav.* 27 (2010), p. 084003. DOI: 10.1088/0264-9381/27/8/084003.
- [48] M Punturo et al. “The Einstein Telescope: a third-generation gravitational wave observatory”. In: *Classical and Quantum Gravity* 27.19 (2010), p. 194002. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/27/19/194002>.
- [49] David Reitze et al. “Cosmic Explorer: the US contribution to gravitational-wave astronomy beyond LIGO”. In: *arXiv preprint arXiv:1907.04833* (2019). URL: <https://arxiv.org/abs/1907.04833>.
- [50] Seiji Kawamura et al. “The Japanese space gravitational wave antenna DECIGO”. In: *Classical and Quantum Gravity* 23.8 (2006), S125. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/23/8/S17>.
- [51] Pau Amaro-Seoane et al. “Laser interferometer space antenna”. In: *arXiv preprint arXiv:1702.00786* (2017).
- [52] R Abbott et al. “Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo”. In: *arXiv preprint arXiv:1912.11716* (2019). URL: <https://arxiv.org/abs/1912.11716>.
- [53] Michele Vallisneri et al. “The LIGO Open Science Center”. In: *Journal of Physics: Conference Series* 610 (May 2015), p. 012021. ISSN: 1742-6596. DOI: 10.1088/1742-6596/610/1/012021. URL: <http://dx.doi.org/10.1088/1742-6596/610/1/012021>.
- [54] Shane L. Larson, William A. Hiscock, and Ronald W. Hellings. “Sensitivity curves for spaceborne gravitational wave interferometers”. In: *Phys. Rev. D* 62 (6 Aug. 2000), p. 062001. DOI: 10.1103/PhysRevD.62.062001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.62.062001>.
- [55] Christopher J Moore, Robert H Cole, and Christopher PL Berry. “Gravitational-wave sensitivity curves”. In: *Classical and Quantum Gravity* 32.1 (2014), p. 015014. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/32/1/015014>.
- [56] Nils Andersson et al. “The transient gravitational-wave sky”. In: *Classical and Quantum Gravity* 30.19 (2013), p. 193002. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/30/19/193002>.
- [57] P. C. Peters. “Gravitational Radiation and the Motion of Two Point Masses”. In: *Phys. Rev.* 136 (4B Nov. 1964), B1224–B1232. DOI: 10.1103/PhysRev.136.B1224. URL: <https://link.aps.org/doi/10.1103/PhysRev.136.B1224>.
- [58] Lee Samuel Finn and David F. Chernoff. “Observing binary inspiral in gravitational radiation: One interferometer”. In: *Phys. Rev. D* 47 (6 Mar. 1993), pp. 2198–2219. DOI: 10.1103/PhysRevD.47.2198. URL: <https://link.aps.org/doi/10.1103/PhysRevD.47.2198>.
- [59] Luc Blanchet. “Gravitational radiation from post-Newtonian sources and inspiralling compact binaries”. In: *Living Reviews in Relativity* 17.1 (2014), p. 2. URL: <https://link.springer.com/article/10.12942/lrr-2014-2>.
- [60] Clifford M Will. “On the unreasonable effectiveness of the post-Newtonian approximation in gravitational physics”. In: *Proceedings of the National Academy of Sciences* 108.15 (2011), pp. 5938–5945. DOI: <https://doi.org/10.1073/pnas.1103127108>.
- [61] Frans Pretorius. “Evolution of Binary Black-Hole Spacetimes”. In: *Phys. Rev. Lett.* 95 (12 Sept. 2005), p. 121101. DOI: 10.1103/PhysRevLett.95.121101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.95.121101>.
- [62] Luca Baiotti and Luciano Rezzolla. “Binary neutron star mergers: a review of Einsteins richest laboratory”. In: *Reports on Progress in Physics* 80.9 (2017), p. 096901. URL: <https://iopscience.iop.org/article/10.1088/1361-6633/aa67bb>.

- [63] Anthony L Piro, Bruno Giacomazzo, and Rosalba Perna. “The fate of neutron star binary mergers”. In: *The Astrophysical Journal Letters* 844.2 (2017), p. L19. URL: <https://iopscience.iop.org/article/10.3847/2041-8213/aa7f2f>.
- [64] B. P. Abbott et al. “Tests of general relativity with GW150914”. In: *Phys. Rev. Lett.* 116 (Feb. 2016), p. 221101. DOI: 10.1103/PhysRevLett.116.221101. arXiv: 1602.03841 [gr-qc].
- [65] Benjamin P Abbott et al. In: *Phys. Rev. Lett.* 123 (1 July 2019), p. 011102. DOI: 10.1103/PhysRevLett.123.011102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.123.011102>.
- [66] Luciano Rezzolla, Elias R Most, and Lukas R Weih. “Using gravitational-wave observations and quasi-universal relations to constrain the maximum mass of neutron stars”. In: *The Astrophysical Journal Letters* 852.2 (2018), p. L25. URL: <https://iopscience.iop.org/article/10.3847/2041-8213/aaa401>.
- [67] M. Isi, M. Pitkin, and A. J. Weinstein. “Probing dynamical gravity with the polarization of continuous gravitational waves”. In: *Phys. Rev. D.* 96.4, 042001 (Aug. 2017), p. 042001. DOI: 10.1103/PhysRevD.96.042001. arXiv: 1703.07530 [gr-qc].
- [68] B. P. Abbott et al. “First Search for Nontensorial Gravitational Waves from Known Pulsars”. In: *Phys. Rev. Lett.* 120.3, 031104 (Jan. 2018), p. 031104. DOI: 10.1103/PhysRevLett.120.031104. arXiv: 1709.09203 [gr-qc].
- [69] Vladimir Dergachev and Maria Alessandra Papa. “Results from the first all-sky search for continuous gravitational waves from small-ellipticity sources”. In: *arXiv preprint arXiv:2004.08334* (2020). URL: <https://arxiv.org/abs/2004.08334>.
- [70] BP Abbott et al. “Searches for gravitational waves from known pulsars at two harmonics in 2015–2017 LIGO data”. In: *The Astrophysical Journal* 879.1 (2019), p. 10. URL: <https://iopscience.iop.org/article/10.3847/1538-4357/ab20cb>.
- [71] James Chadwick. “Possible existence of a neutron”. In: *Nature* 129.3252 (1932), pp. 312–312.
- [72] W. Baade and F. Zwicky. “Remarks on Super-Novae and Cosmic Rays”. In: *Phys. Rev.* 46 (1 July 1934), pp. 76–77. DOI: 10.1103/PhysRev.46.76.2. URL: <https://link.aps.org/doi/10.1103/PhysRev.46.76.2>.
- [73] Franco Pacini. “Energy emission from a neutron star”. In: *Nature* 216.5115 (1967), pp. 567–568.
- [74] Antony Hewish et al. “Observation of a rapidly pulsating radio source”. In: *Nature* 217.5130 (1968), pp. 709–713.
- [75] J. R. Oppenheimer and G. M. Volkoff. “On Massive Neutron Cores”. In: *Phys. Rev.* 55 (4 Feb. 1939), pp. 374–381. DOI: 10.1103/PhysRev.55.374. URL: <https://link.aps.org/doi/10.1103/PhysRev.55.374>.
- [76] S. Typel et al. “Composition and thermodynamics of nuclear matter with light clusters”. In: *Phys. Rev. C* 81 (1 Jan. 2010), p. 015803. DOI: 10.1103/PhysRevC.81.015803. URL: <https://link.aps.org/doi/10.1103/PhysRevC.81.015803>.
- [77] H Thankful Cromartie et al. “Relativistic Shapiro delay measurements of an extremely massive millisecond pulsar”. In: *Nature Astronomy* 4.1 (2020), pp. 72–76. URL: <https://www.nature.com/articles/s41550-019-0880-2>.
- [78] Max Camenzind. *Compact objects in astrophysics: white dwarfs, neutron stars and black holes*. Springer Science & Business Media, 2007.
- [79] Pawel Haensel, Aleksander Yu Potekhin, and Dmitry G Yakovlev. *Neutron stars 1: Equation of state and structure*. Vol. 326. Springer Science & Business Media, 2007.

A. Bibliography

- [80] Mathieu Servillat et al. “Neutron star atmosphere composition: the quiescent, low-mass X-ray binary in the globular cluster M28”. In: *Monthly Notices of the Royal Astronomical Society* 423.2 (2012), pp. 1556–1561. URL: <https://academic.oup.com/mnras/article/423/2/1556/963431>.
- [81] Wynn CG Ho and Craig O Heinke. “A neutron star with a carbon atmosphere in the Cassiopeia A supernova remnant”. In: *Nature* 462.7269 (2009), pp. 71–73. URL: <https://www.nature.com/articles/nature08525>.
- [82] B. P. Abbott et al. “GW170817: Measurements of Neutron Star Radii and Equation of State”. In: *Phys. Rev. Lett.* 121 (16 Oct. 2018), p. 161101. DOI: 10.1103/PhysRevLett.121.161101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.121.161101>.
- [83] M. Zimmermann and E. Szedenits Jr. “Gravitational waves from rotating and precessing rigid bodies - Simple models and applications to pulsars”. In: *Phys. Rev. D.* 20 (July 1979), pp. 351–355. DOI: 10.1103/PhysRevD.20.351.
- [84] M. Zimmermann. “Gravitational waves from rotating and precessing rigid bodies. II - General solutions and computationally useful formulas”. In: *Phys. Rev. D.* 21 (Feb. 1980), pp. 891–898. DOI: 10.1103/PhysRevD.21.891.
- [85] P. D. Lasky. “Gravitational Waves from Neutron Stars: A Review”. In: *PASA* 32, e034 (Sept. 2015), e034. DOI: 10.1017/pasa.2015.35. arXiv: 1508.06643 [astro-ph.HE].
- [86] Lee Lindblom, Benjamin J. Owen, and Sharon M. Morsink. “Gravitational Radiation Instability in Hot Young Neutron Stars”. In: *Phys. Rev. Lett.* 80 (22 June 1998), pp. 4843–4846. DOI: 10.1103/PhysRevLett.80.4843. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.80.4843>.
- [87] Benjamin J. Owen et al. “Gravitational waves from hot young rapidly rotating neutron stars”. In: *Phys. Rev. D* 58 (8 Sept. 1998), p. 084020. DOI: 10.1103/PhysRevD.58.084020. URL: <https://link.aps.org/doi/10.1103/PhysRevD.58.084020>.
- [88] Santiago Caride et al. “How to search for gravitational waves from r -modes of known pulsars”. In: *Phys. Rev. D* 100 (6 Sept. 2019), p. 064013. DOI: 10.1103/PhysRevD.100.064013. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.064013>.
- [89] Maximiliano Isi et al. “Directed searches for gravitational waves from ultralight bosons”. In: *Phys. Rev. D* 99 (8 Apr. 2019), p. 084042. DOI: 10.1103/PhysRevD.99.084042. URL: <https://link.aps.org/doi/10.1103/PhysRevD.99.084042>.
- [90] Asimina Arvanitaki et al. “String axiverse”. In: *Phys. Rev. D* 81 (12 June 2010), p. 123530. DOI: 10.1103/PhysRevD.81.123530. URL: <https://link.aps.org/doi/10.1103/PhysRevD.81.123530>.
- [91] K. Riles. “Recent searches for continuous gravitational waves”. In: *Modern Physics Letters A* 32, 1730035-685 (Dec. 2017), pp. 1730035–685. DOI: 10.1142/S021773231730035X. arXiv: 1712.05897 [gr-qc].
- [92] Asimina Arvanitaki, Masha Baryakhtar, and Xinlu Huang. “Discovering the QCD axion with black holes and gravitational waves”. In: *Phys. Rev. D* 91 (8 Apr. 2015), p. 084011. DOI: 10.1103/PhysRevD.91.084011. URL: <https://link.aps.org/doi/10.1103/PhysRevD.91.084011>.
- [93] C. Palomba et al. “Direct Constraints on the Ultralight Boson Mass from Searches of Continuous Gravitational Waves”. In: *Phys. Rev. Lett.* 123 (17 Oct. 2019), p. 171101. DOI: 10.1103/PhysRevLett.123.171101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.123.171101>.
- [94] Sylvia J Zhu et al. “Characterizing the continuous gravitational-wave signal from boson clouds around Galactic isolated black holes”. In: *arXiv preprint arXiv:2003.03359* (2020). URL: <https://arxiv.org/abs/2003.03359>.

- [95] Gijs Nelemans, LR Yungelson, and SF Portegies Zwart. “The gravitational wave signal from the Galactic disk population of binaries containing two compact objects”. In: *Astronomy & Astrophysics* 375.3 (2001), pp. 890–898. URL: <https://www.aanda.org/articles/aa/abs/2001/33/aah2754/aah2754.html>.
- [96] John T Whelan, Reinhard Prix, and Deepak Khurana. “Improved search for galactic white-dwarf binaries in mock lisa data challenge 1b using an-statistic template bank”. In: *Classical and Quantum Gravity* 25.18 (2008), p. 184029. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/25/18/184029>.
- [97] Surajit Kalita and Banibrata Mukhopadhyay. “Continuous gravitational wave from magnetized white dwarfs and neutron stars: possible missions for LISA, DECIGO, BBO, ET detectors”. In: *Monthly Notices of the Royal Astronomical Society* 490.2 (Oct. 2019), pp. 2692–2705. ISSN: 0035-8711. DOI: 10.1093/mnras/stz2734. eprint: <https://academic.oup.com/mnras/article-pdf/490/2/2692/30296776/stz2734.pdf>. URL: <https://doi.org/10.1093/mnras/stz2734>.
- [98] J. H. Taylor and J. M. Weisberg. “Further Experimental Tests of Relativistic Gravity Using the Binary Pulsar PSR 1913+16”. In: *ApJ* 345 (Oct. 1989), p. 434. DOI: 10.1086/167917.
- [99] Kai Liao, Marek Biesiada, and Xi-Long Fan. “The wave nature of continuous gravitational waves from microlensing”. In: *The Astrophysical Journal* 875.2 (2019), p. 139. URL: <https://iopscience.iop.org/article/10.3847/1538-4357/ab1087>.
- [100] E Goetz and K Riles. “An all-sky search algorithm for continuous gravitational waves from spinning neutron stars in binary systems”. In: *Class. Quant. Grav.* 28.21 (2011), p. 215006. DOI: 10.1088/0264-9381/28/21/215006.
- [101] P. B. Covas and Alicia M. Sintes. “New method to search for continuous gravitational waves from unknown neutron stars in binary systems”. In: *Phys. Rev. D* 99 (12 June 2019), p. 124019. DOI: 10.1103/PhysRevD.99.124019. URL: <https://link.aps.org/doi/10.1103/PhysRevD.99.124019>.
- [102] Piotr Jaranowski, Andrzej Krolak, and Bernard F Schutz. “Data analysis of gravitational-wave signals from spinning neutron stars: The signal and its detection”. In: *Phys. Rev. D.* 58.6 (1998), p. 063001. DOI: 10.1103/PhysRevD.58.063001.
- [103] Louis J. Rubbo, Neil J. Cornish, and Olivier Poujade. “Forward modeling of space-borne gravitational wave detectors”. In: *Phys. Rev. D* 69 (8 Apr. 2004), p. 082003. DOI: 10.1103/PhysRevD.69.082003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.69.082003>.
- [104] Bernard F. Schutz and Massimo Tinto. “Antenna patterns of interferometric detectors of gravitational waves ? I. Linearly polarized waves”. In: *Monthly Notices of the Royal Astronomical Society* 224.1 (Jan. 1987), pp. 131–154. ISSN: 0035-8711. DOI: 10.1093/mnras/224.1.131. eprint: <https://academic.oup.com/mnras/article-pdf/224/1/131/4099535/mnras224-0131.pdf>. URL: <https://doi.org/10.1093/mnras/224.1.131>.
- [105] J. T. Whelan et al. “New coordinates for the amplitude parameter space of continuous gravitational waves”. In: *Class. Quant. Grav.* 31.6 (Mar. 2014), p. 065002. DOI: 10.1088/0264-9381/31/6/065002. eprint: 1311.0065.
- [106] Reinhard Prix. “The F-statistic and its implementation in ComputeFStatistic v2”. In: *LIGO-T0900149* (2010). URL: <https://dcc.ligo.org/LIGO-T0900149/public>.

A. Bibliography

- [107] Jerzy Neyman and Egon Sharpe Pearson. “IX. On the problem of the most efficient tests of statistical hypotheses”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231.694-706 (1933), pp. 289–337.
- [108] R. Prix and B. Krishnan. “Targeted search for continuous gravitational waves: Bayesian versus maximum-likelihood statistics”. In: *Class. Quant. Grav.* 26.20 (Oct. 2009), pp. 204013–+. DOI: 10.1088/0264-9381/26/20/204013. eprint: 0907.2569.
- [109] A. C. Searle. “Monte-Carlo and Bayesian techniques in gravitational wave burst data analysis”. In: *ArXiv e-prints* (Apr. 2008). arXiv: 0804.1161 [gr-qc].
- [110] Reinhard Prix, Stefanos Giampanis, and Chris Messenger. “Search method for long-duration gravitational-wave transients from neutron stars”. In: *Phys. Rev. D* 84 (2011). (eprint arXiv:1104.1704), p. 023007. eprint: arXiv:1104.1704.
- [111] Nelson Christensen et al. “Metropolis-Hastings algorithm for extracting periodic gravitational wave signals from laser interferometric detector data”. In: *Phys. Rev. D* 70 (2 July 2004), p. 022001. DOI: 10.1103/PhysRevD.70.022001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.70.022001>.
- [112] R. J. Dupuis and G. Woan. “Bayesian estimation of pulsar parameters from gravitational wave data”. In: *Phys. Rev. D* 72.10, 102002 (Nov. 2005), p. 102002. DOI: 10.1103/PhysRevD.72.102002. eprint: gr-qc/0508096.
- [113] Bernard F Schutz. “Sources of radiation from neutron stars”. In: *arXiv preprint gr-qc/9802020* (1998). URL: <https://arxiv.org/abs/gr-qc/9802020>.
- [114] Bernard F Schutz and M Alessandra Papa. “End-to-end algorithm for hierarchical area searches for long-duration GW sources for GEO 600”. In: *arXiv preprint gr-qc/9905018* (1999). URL: <https://arxiv.org/abs/gr-qc/9905018>.
- [115] Peter R Williams and Bernard F Schutz. “An efficient matched filtering algorithm for the detection of continuous gravitational wave signals”. In: *AIP Conference Proceedings*. Vol. 523. 1. American Institute of Physics. 2000, pp. 473–476. DOI: <https://doi.org/10.1063/1.1291918>.
- [116] Lee S. Finn. “Detection, measurement, and gravitational radiation”. In: *Phys. Rev. D* 46 (1992), pp. 5236–5249. DOI: 10.1103/PhysRevD.46.5236.
- [117] C. Cutler and B. F. Schutz. “Generalized F-statistic: Multiple detectors and multiple gravitational wave pulsars”. In: *Phys. Rev. D* 72.6, 063006 (Sept. 2005), p. 063006. DOI: 10.1103/PhysRevD.72.063006. eprint: gr-qc/0504011.
- [118] R. Prix. “Search for continuous gravitational waves: Metric of the multidetector F-statistic”. In: *Phys. Rev. D* 75.2, 023004 (Jan. 2007), p. 023004. DOI: 10.1103/PhysRevD.75.023004. eprint: gr-qc/0606088.
- [119] Reinhard Prix. personal communication. June 11, 2020.
- [120] Matthew Pitkin et al. “A new code for parameter estimation in searches for gravitational waves from known pulsars”. In: *Journal of Physics: Conference Series*. Vol. 363. 1. IOP Publishing. 2012, p. 012041. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/363/1/012041>.
- [121] G. Ashton and R. Prix. “Hierarchical multistage MCMC follow-up of continuous gravitational wave candidates”. In: *Phys. Rev. D* 97 (10 May 2018), p. 103020. DOI: 10.1103/PhysRevD.97.103020. URL: <https://link.aps.org/doi/10.1103/PhysRevD.97.103020>.

- [122] R. Balasubramanian, B. S. Sathyaprakash, and S. V. Dhurandhar. “Gravitational waves from coalescing binaries: Detection strategies and Monte Carlo estimation of parameters”. In: *Phys. Rev. D* 53 (6 Mar. 1996), pp. 3033–3055. DOI: 10.1103/PhysRevD.53.3033. URL: <https://link.aps.org/doi/10.1103/PhysRevD.53.3033>.
- [123] Benjamin J. Owen. “Search templates for gravitational waves from inspiraling binaries: Choice of template spacing”. In: *Phys. Rev. D* 53 (12 June 1996), pp. 6749–6761. DOI: 10.1103/PhysRevD.53.6749. URL: <https://link.aps.org/doi/10.1103/PhysRevD.53.6749>.
- [124] Karl Wette and Reinhard Prix. “Flat parameter-space metric for all-sky searches for gravitational-wave pulsars”. In: *Phys. Rev. D* 88 (12 Dec. 2013), p. 123005. DOI: 10.1103/PhysRevD.88.123005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.88.123005>.
- [125] BP Abbott et al. “Searches for continuous gravitational waves from 15 supernova remnants and Fomalhaut b with advanced LIGO”. In: *The Astrophysical Journal* 875.2 (2019), p. 122. URL: <https://iopscience.iop.org/article/10.3847/1538-4357/ab113b>.
- [126] P. R. Brady and T. Creighton. “Searching for periodic sources with LIGO. II. Hierarchical searches”. In: *Phys. Rev. D*. 61.8, 082001 (Apr. 2000), p. 082001. DOI: 10.1103/PhysRevD.61.082001. eprint: [gr-qc/9812014](https://arxiv.org/abs/gr-qc/9812014).
- [127] The LIGO Scientific Collaboration et al. “Full Band All-sky Search for Periodic Gravitational Waves in the O1 LIGO Data”. In: *Phys. Rev. D*. 97 (2018), p. 102003. DOI: 10.1103/PhysRevD.97.102003. arXiv: 1802.05241 [gr-qc].
- [128] M Alessandra Papa et al. “Search for Continuous Gravitational Waves from the Central Compact Objects in Supernova Remnants Cassiopeia A, Vela Jr. and G347. 3-0.5”. In: *arXiv* (2020), arXiv-2005. URL: <https://arxiv.org/abs/2005.06544>.
- [129] C. Cutler, I. Gholami, and B. Krishnan. “Improved stack-slide searches for gravitational-wave pulsars”. In: *Phys. Rev. D*. 72 (Aug. 2005), p. 042004. DOI: 10.1103/PhysRevD.72.042004. eprint: [gr-qc/0505082](https://arxiv.org/abs/gr-qc/0505082).
- [130] B. Krishnan et al. “Hough transform search for continuous gravitational waves”. In: *Phys. Rev. D*. 70.8, 082001 (Oct. 2004), p. 082001. DOI: 10.1103/PhysRevD.70.082001. eprint: [gr-qc/0407001](https://arxiv.org/abs/gr-qc/0407001).
- [131] Vladimir Dergachev. “Description of PowerFlux algorithms and implementation”. In: *LIGO-T050186* (2005).
- [132] Vladimir Dergachev. “Description of PowerFlux 2 algorithms and implementation”. In: *LIGO-T1000272* (2010). URL: <https://dcc.ligo.org/LIGO-T1000272/public>.
- [133] Vladimir Dergachev and Maria Alessandra Papa. “Sensitivity Improvements in the Search for Periodic Gravitational Waves Using O1 LIGO Data”. In: *Phys. Rev. Lett.* 123 (10 Sept. 2019), p. 101101. DOI: 10.1103/PhysRevLett.123.101101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.123.101101>.
- [134] Sanjeev Dhurandhar et al. “Cross-correlation search for periodic gravitational waves”. In: *Phys. Rev. D*. 77.8 (2008), p. 082001. DOI: 10.1103/PhysRevD.77.082001.
- [135] B. P. Abbott et al. “Upper Limits on Gravitational Waves from Scorpius X-1 from a Model-based Cross-correlation Search in Advanced LIGO Data”. In: *ApJ* 847, 47 (Sept. 2017), p. 47. DOI: 10.3847/1538-4357/aa86f0.
- [136] S. Suvorova et al. “Hidden Markov model tracking of continuous gravitational waves from a neutron star with wandering spin”. In: *Phys. Rev. D* 93 (12 June 2016), p. 123009. DOI: 10.1103/PhysRevD.93.123009. URL: <https://link.aps.org/doi/10.1103/PhysRevD.93.123009>.

A. Bibliography

- [137] Joe Bayley, Chris Messenger, and Graham Woan. “Generalized application of the Viterbi algorithm to searches for continuous gravitational-wave signals”. In: *Phys. Rev. D* 100 (2 July 2019), p. 023006. DOI: 10.1103/PhysRevD.100.023006. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.023006>.
- [138] H. J. Pletsch and B. Allen. “Exploiting Large-Scale Correlations to Detect Continuous Gravitational Waves”. In: *Phys. Rev. Lett.* 103.18 (Oct. 2009), pp. 181102–+. DOI: 10.1103/PhysRevLett.103.181102. eprint: 0906.0023.
- [139] Reinhard Prix Miroslav Shaltev. “Fully coherent follow-up of continuous gravitational-wave candidates”. In: *Phys. Rev. D.* 87 (2013), p. 084057. DOI: 10.1103/PhysRevD.87.084057. eprint: arXiv:1303.2471.
- [140] Derek Davis et al. “Improving the sensitivity of Advanced LIGO using noise subtraction”. In: *Classical and Quantum Gravity* 36.5 (2019), p. 055011.
- [141] S. J. Zhu, M. A. Papa, and S. Walsh. “New veto for continuous gravitational wave searches”. In: *Phys. Rev. D.* 96.12, 124007 (Dec. 2017), p. 124007. DOI: 10.1103/PhysRevD.96.124007. arXiv: 1707.05268 [gr-qc].
- [142] David Keitel et al. “Search for continuous gravitational waves: Improving robustness versus instrumental artifacts”. In: *Phys. Rev. D* 89 (6 Mar. 2014), p. 064023. DOI: 10.1103/PhysRevD.89.064023. URL: <https://link.aps.org/doi/10.1103/PhysRevD.89.064023>.
- [143] David Keitel and Reinhard Prix. “Line-robust statistics for continuous gravitational waves: safety in the case of unequal detector sensitivities”. In: *Classical and Quantum Gravity* 32.3 (2015), p. 035004. URL: <https://iopscience.iop.org/article/10.1088/0264-9381/32/3/035004>.
- [144] Avneet Singh et al. “Adaptive clustering procedure for continuous gravitational wave searches”. In: *Phys. Rev. D* 96 (8 Oct. 2017), p. 082003. DOI: 10.1103/PhysRevD.96.082003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.96.082003>.
- [145] K. Wette et al. “Weave: a semicoherent search implementation for continuous gravitational waves”. In: *ArXiv e-prints* (Apr. 2018). arXiv: 1804.03392 [astro-ph.IM].
- [146] Nils J Nilsson. *The quest for artificial intelligence*. Cambridge University Press, 2009.
- [147] Chollet François. *Deep learning with Python*. Manning Publications Company, 2017.
- [148] Linda K Woolery and Jerzy Grzymala-Busse. “Machine learning for an expert system to predict preterm birth risk”. In: *Journal of the American Medical Informatics Association* 1.6 (1994), pp. 439–446. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC116227/>.
- [149] Arthur N Rasmussen. “The INCO Expert System Project: CLIPS in Shuttle Mission Control”. In: *First CLIPSConference* (1990), p. 305.
- [150] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2019.
- [151] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [152] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. URL: <http://neuralnetworksanddeeplearning.com/>.
- [153] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. “Multilayer feedforward networks are universal approximators.” In: *Neural networks* 2.5 (1989), pp. 359–366. DOI: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8).
- [154] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314. DOI: <https://doi.org/10.1007/BF02551274>.

- [155] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: <http://www.deeplearningbook.org/>.
- [156] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536. DOI: <https://doi.org/10.1038/323533a0>.
- [157] Richard Hahnloser et al. “Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex”. In: *Nature* (Jan. 2000). URL: <https://www.nature.com/articles/35016072>.
- [158] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feed-forward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 249–256. URL: <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf>.
- [159] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml*. Vol. 30. 1. 2013, p. 3. URL: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.
- [160] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015). URL: <https://arxiv.org/abs/1511.07289>.
- [161] Günter Klambauer et al. “Self-normalizing neural networks”. In: *Advances in neural information processing systems*. 2017, pp. 971–980. URL: <https://arxiv.org/abs/1706.02515>.
- [162] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014). URL: <https://arxiv.org/abs/1412.6980>.
- [163] Timothy Dozat. “Incorporating nesterov momentum into adam”. In: (2016). URL: <https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ>.
- [164] Matthew D Zeiler. “ADADELTA: an adaptive learning rate method”. In: *arXiv preprint arXiv:1212.5701* (2012). URL: <https://arxiv.org/abs/1212.5701>.
- [165] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization.” In: *Journal of machine learning research* 12.7 (2011). URL: https://stanford.edu/~jduchi/projects/DuchiHaSi10_colt.pdf.
- [166] John Hertz et al. “Introduction to the theory of neural computation”. In: *Physics Today* 44 (1991), p. 70.
- [167] Geoffrey E Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint arXiv:1207.0580* (2012). URL: <https://arxiv.org/abs/1207.0580>.
- [168] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks”. In: *International conference on machine learning*. 2013, pp. 1310–1318. URL: <https://arxiv.org/abs/1211.5063>.
- [169] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105. URL: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [170] K. He et al. “Deep Residual Learning for Image Recognition”. In: *arXiv e-prints* (Dec. 2015). arXiv: 1512.03385 [cs.CV].

A. Bibliography

- [171] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning, Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, pp. 448–456. URL: <https://arxiv.org/abs/1502.03167>.
- [172] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*. 2009. URL: http://www.image-net.org/papers/imagenet_cvpr09.pdf.
- [173] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9. URL: <https://arxiv.org/abs/1409.4842>.
- [174] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.
- [175] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017. URL: <https://arxiv.org/pdf/1602.07261.pdf>.
- [176] B. P. Abbott et al. “Binary Black Hole Mergers in the first Advanced LIGO Observing Run”. In: *Phys. Rev. X*. 6.41015 (June 2016). DOI: 10.1103/PhysRevX.6.041015. arXiv: 1606.04856 [gr-qc].
- [177] Reinhard Prix. “Gravitational waves from spinning neutron stars”. In: *Neutron Stars and Pulsars*. Springer, 2009, pp. 651–685. URL: <https://dcc.ligo.org/LIGO-P060039/public>.
- [178] N. K. Johnson-McDaniel and B. J. Owen. “Maximum elastic deformations of relativistic stars”. In: *Phys. Rev. D*. 88.4, 044004 (Aug. 2013), p. 044004. DOI: 10.1103/PhysRevD.88.044004. arXiv: 1208.5227 [astro-ph.SR].
- [179] B. Knispel and B. Allen. “Blandford’s argument: The strongest continuous gravitational wave signal”. In: *Phys. Rev. D*. 78.4 (Aug. 2008), p. 044031. DOI: 10.1103/PhysRevD.78.044031. eprint: 0804.3075.
- [180] R. Prix and M. Shaltev. “Search for continuous gravitational waves: Optimal StackSlide method at fixed computing cost”. In: *Phys. Rev. D*. 85.8, 084010 (Apr. 2012), p. 084010. DOI: 10.1103/PhysRevD.85.084010. arXiv: 1201.4321 [gr-qc].
- [181] R. N. Manchester et al. “The Australia telescope national facility pulsar catalogue”. In: *The Astronomical Journal* 129.4 (2005), p. 1993. URL: <http://www.atnf.csiro.au/research/pulsar/psrcat>.
- [182] B. P. Abbott et al. “First Search for Gravitational Waves from Known Pulsars with Advanced LIGO”. In: *ApJ* 839, 12 (Apr. 2017), p. 12. DOI: 10.3847/1538-4357/aa677f. arXiv: 1701.07709 [astro-ph.HE].
- [183] B. P. Abbott et al. “First narrow-band search for continuous gravitational waves from known pulsars in advanced detector data”. In: *Phys. Rev. D*. 96.12, 122006 (Dec. 2017), p. 122006. DOI: 10.1103/PhysRevD.96.122006.
- [184] J. Aasi et al. “Searches for Continuous Gravitational Waves from Nine Young Supernova Remnants”. In: *ApJ* 813, 39 (Nov. 2015), p. 39. DOI: 10.1088/0004-637X/813/1/39. arXiv: 1412.5942 [astro-ph.HE].
- [185] Sylvia J Zhu et al. “Einstein@Home search for continuous gravitational waves from Cassiopeia A”. In: *Phys. Rev. D*. 94.8 (2016), p. 082008. DOI: 10.1103/PhysRevD.94.082008.
- [186] B. P. Abbott et al. “Search for gravitational waves from Scorpius X-1 in the first Advanced LIGO observing run with a hidden Markov model”. In: *Phys. Rev. D*. 95.12, 122003 (June 2017), p. 122003. DOI: 10.1103/PhysRevD.95.122003. arXiv: 1704.03719 [gr-qc].

- [187] Benjamin P Abbott et al. “All-sky search for periodic gravitational waves in the O1 LIGO data”. In: *Phys. Rev. D.* 96.6 (2017), p. 062002. DOI: 10.1103/PhysRevD.96.062002.
- [188] BP Abbott et al. “First low-frequency Einstein@Home all-sky search for continuous gravitational waves in Advanced LIGO data”. In: *Phys. Rev. D.* 96 (2017), p. 122004. DOI: 10.1103/PhysRevD.96.122004.
- [189] J Aasi et al. “First all-sky search for continuous gravitational waves from unknown sources in binary systems”. In: *Phys. Rev. D.* 90.6 (2014), p. 062010. DOI: 10.1103/PhysRevD.90.062010.
- [190] J. Ming et al. “Optimal directed searches for continuous gravitational waves”. In: *Phys. Rev. D.* 93.6, 064011 (Mar. 2016), p. 064011. DOI: 10.1103/PhysRevD.93.064011. arXiv: 1510.03417 [gr-qc].
- [191] Jing Ming et al. “Optimally setting up directed searches for continuous gravitational waves in Advanced LIGO O1 data”. In: *Phys. Rev. D.* 97, 024051 (Feb. 2018), p. 024051. DOI: 10.1103/PhysRevD.97.024051.
- [192] P. Leaci and R. Prix. “Directed searches for continuous gravitational waves from binary systems: parameter-space metrics and optimal Scorpius X-1 sensitivity”. In: *Phys. Rev. D.* 91 (Feb. 2015), p. 102003. eprint: 1502.00914.
- [193] B. Abbott et al. “Setting upper limits on the strength of periodic gravitational waves from PSR J1939+2134 using the first science data from the GEO 600 and LIGO detectors”. In: *Phys. Rev. D.* 69.8, 082004 (Apr. 2004), p. 082004. DOI: 10.1103/PhysRevD.69.082004. eprint: gr-qc/0308050.
- [194] G. Mendell and M. Landry. *StackSlide and Hough Search SNR and Statistics*. Tech. rep. LIGO Technical Document. 2005. eprint: LIGO-T050003-x0. URL: <https://dcc.ligo.org/LIGO-T050003/public>.
- [195] B Abbott et al. “All-sky search for periodic gravitational waves in LIGO S4 data”. In: *Phys. Rev. D.* 77.2 (2008), p. 022001. DOI: 10.1103/PhysRevD.77.022001.
- [196] Junaid Aasi et al. “Einstein@Home all-sky search for periodic gravitational waves in LIGO S5 data”. In: *Phys. Rev. D.* 87.4 (2013), p. 042001. DOI: 10.1103/PhysRevD.87.042001.
- [197] Reinhard Prix. *The \mathcal{F} -statistic and its implementation in ComputeFStatistic_v2*. Tech. rep. (LIGO-T0900149-v5). 2015. URL: <https://dcc.ligo.org/LIGO-T0900149/public>.
- [198] K. Wette. “Parameter-space metric for all-sky semicoherent searches for gravitational-wave pulsars”. In: *Phys. Rev. D.* 92.8, 082003 (Oct. 2015), p. 082003. DOI: 10.1103/PhysRevD.92.082003. arXiv: 1508.02372 [gr-qc].
- [199] B. Behnke, M. A. Papa, and R. Prix. “Postprocessing methods used in the search for continuous gravitational-wave signals from the Galactic Center”. In: *Phys. Rev. D.* 91.6, 064007 (Mar. 2015), p. 064007. DOI: 10.1103/PhysRevD.91.064007. arXiv: 1410.5997 [gr-qc].
- [200] J. Abadie et al. “First Search for Gravitational Waves from the Youngest Known Neutron Star”. In: *ApJ* 722 (Oct. 2010), pp. 1504–1513. DOI: 10.1088/0004-637X/722/2/1504. arXiv: 1006.2535 [gr-qc].
- [201] M. A. Papa et al. “Hierarchical follow-up of subthreshold candidates of an all-sky Einstein@Home search for continuous gravitational waves on LIGO sixth science run data”. In: *Phys. Rev. D.* 94.12, 122006 (Dec. 2016), p. 122006. DOI: 10.1103/PhysRevD.94.122006.
- [202] C. Röver, C. Messenger, and R. Prix. “Bayesian versus frequentist upper limits”. In: *Proceedings of the PHYSTAT 2011 Workshop*. Mar. 2011. arXiv: 1103.2987 [physics.data-an].

A. Bibliography

- [203] J. Aasi et al. “Gravitational Waves from Known Pulsars: Results from the Initial Detector Era”. In: *ApJ* 785, 119 (Apr. 2014), p. 119. DOI: 10.1088/0004-637X/785/2/119. arXiv: 1309.4027 [astro-ph.HE].
- [204] John W. Eaton et al. *GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations*. 2015. URL: <http://www.gnu.org/software/octave/doc/interpreter>.
- [205] Karl Wette et al. “OctApps: a library of Octave functions for continuous gravitational-wave data analysis”. In: *Journal of Open Source Software* 3.26 (2018), p. 707. DOI: 10.21105/joss.00707.
- [206] Reinhard Prix and Karl Wette. “Estimating sensitivity of the Einstein@Home search S5R5”. In: *LIGO-T1200272* (2012). URL: <https://dcc.ligo.org/LIGO-T1200272/public>.
- [207] B. P. Abbott et al. “Results of the deepest all-sky survey for continuous gravitational waves on LIGO S6 data running on the Einstein@Home volunteer distributed computing project”. In: *Phys. Rev. D*. 94.10, 102002 (Nov. 2016), p. 102002. DOI: 10.1103/PhysRevD.94.102002. arXiv: 1606.09619 [gr-qc].
- [208] Peter J. Rousseeuw and Christophe Croux. “Alternatives to the Median Absolute Deviation”. In: *Journal of the American Statistical Association* 88.424 (1993), pp. 1273–1283. DOI: 10.1080/01621459.1993.10476408. eprint: <https://amstat.tandfonline.com/doi/pdf/10.1080/01621459.1993.10476408>. URL: <https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1993.10476408>.
- [209] B Abbott et al. “First all-sky upper limits from LIGO on the strength of periodic gravitational waves using the Hough transform”. In: *Phys. Rev. D*. 72.10 (2005), p. 102004. DOI: 10.1103/PhysRevD.72.102004.
- [210] B. Abbott et al. “Searches for periodic gravitational waves from unknown isolated sources and Scorpius X-1: Results from the second LIGO science run”. In: *Phys. Rev. D*. 76.8, 082001 (Oct. 2007), p. 082001. DOI: 10.1103/PhysRevD.76.082001. eprint: [gr-qc/0605028](https://arxiv.org/abs/gr-qc/0605028).
- [211] B Abbott et al. “Einstein@Home search for periodic gravitational waves in LIGO S4 data”. In: *Phys. Rev. D*. 79.2 (2009), p. 022001. DOI: 10.1103/PhysRevD.79.022001.
- [212] B. P. Abbott et al. “All-Sky LIGO Search for Periodic Gravitational Waves in the Early Fifth-Science-Run Data”. In: *Phys. Rev. Lett.* 102.11 (2009), p. 111102. DOI: 10.1103/PhysRevLett.102.111102. eprint: [0810.0283](https://arxiv.org/abs/0810.0283).
- [213] BP Abbott et al. “Einstein@Home search for periodic gravitational waves in early S5 LIGO data”. In: *Phys. Rev. D*. 80.4 (2009), p. 042003. DOI: 10.1103/PhysRevD.80.042003.
- [214] J Abadie et al. “All-sky search for periodic gravitational waves in the full S5 LIGO data”. In: *Phys. Rev. D*. 85.2 (2012), p. 022001. DOI: 10.1103/PhysRevD.85.022001.
- [215] J Aasi et al. “Application of a Hough search for continuous gravitational waves on data from the fifth LIGO science run”. In: *Class. Quant. Grav.* 31.8 (2014), p. 085014. DOI: 10.1088/0264-9381/31/8/085014.
- [216] A. Singh et al. “Results of an all-sky high-frequency Einstein@Home search for continuous gravitational waves in LIGO’s fifth science run”. In: *Phys. Rev. D*. 94.6, 064061 (Sept. 2016), p. 064061. DOI: 10.1103/PhysRevD.94.064061. arXiv: 1607.00745 [gr-qc].
- [217] J. Aasi et al. “Implementation of an F-statistic all-sky search for continuous gravitational waves in Virgo VSR1 data”. In: *Class. Quant. Grav.* 31.16, 165014 (Aug. 2014), p. 165014. DOI: 10.1088/0264-9381/31/16/165014. arXiv: 1402.4974 [gr-qc].

- [218] J Aasi et al. “First low frequency all-sky search for continuous gravitational wave signals”. In: *Phys. Rev. D.* 93.4 (2016), p. 042007. DOI: 10.1103/PhysRevD.93.042007.
- [219] BP Abbott et al. “Comprehensive all-sky search for periodic gravitational waves in the sixth science run LIGO data”. In: *Phys. Rev. D.* 94.4 (2016), p. 042002. DOI: 10.1103/PhysRevD.94.042002.
- [220] Junaid Aasi et al. “Directed search for continuous gravitational waves from the Galactic center”. In: *Phys. Rev. D.* 88.10 (2013), p. 102002. DOI: 10.1103/PhysRevD.88.102002.
- [221] B Abbott et al. “Beating the spin-down limit on gravitational wave emission from the Crab pulsar”. In: *ApJL* 683.1 (2008), p. L45. DOI: 10.1086/591526.
- [222] J. Aasi et al. “Narrow-band search of continuous gravitational-wave signals from Crab and Vela pulsars in Virgo VSR4 data”. In: *Phys. Rev. D.* 91.2, 022004 (Jan. 2015), p. 022004. DOI: 10.1103/PhysRevD.91.022004. arXiv: 1410.8310 [astro-ph.IM].
- [223] J. Aasi et al. “Search of the Orion spur for continuous gravitational waves using a loosely coherent algorithm on data from LIGO interferometers”. In: *Phys. Rev. D.* 93.4, 042006 (Feb. 2016), p. 042006. DOI: 10.1103/PhysRevD.93.042006. arXiv: 1510.03474 [gr-qc].
- [224] B. P. Abbott et al. “Search for continuous gravitational waves from neutron stars in globular cluster NGC 6544”. In: *Phys. Rev. D.* 95.8, 082005 (Apr. 2017), p. 082005. DOI: 10.1103/PhysRevD.95.082005. arXiv: 1607.02216 [gr-qc].
- [225] B. P. Abbott et al. “Directional Limits on Persistent Gravitational Waves from Advanced LIGO’s First Observing Run”. In: *Phys. Rev. Lett.* 118.12, 121102 (Mar. 2017), p. 121102. DOI: 10.1103/PhysRevLett.118.121102. arXiv: 1612.02030 [gr-qc].
- [226] J Aasi et al. “A directed search for gravitational waves from Scorpius X-1 with initial LIGO data”. In: *Phys. Rev. D.* 91.6 (2015), p. 062008. DOI: 10.1103/PhysRevD.91.062008.
- [227] G. D. Meadors et al. “Searches for continuous gravitational waves from Scorpius X-1 and XTE J1751-305 in LIGO’s sixth science run”. In: *Phys. Rev. D.* 95.4, 042005 (Feb. 2017), p. 042005. DOI: 10.1103/PhysRevD.95.042005. arXiv: 1610.09391 [gr-qc].
- [228] B. Abbott et al. “Limits on Gravitational-Wave Emission from Selected Pulsars Using LIGO Data”. In: *Phys. Rev. Lett.* 94.18, 181103 (May 2005), p. 181103. DOI: 10.1103/PhysRevLett.94.181103. eprint: gr-qc/0410007.
- [229] B. Abbott et al. “Upper limits on gravitational wave emission from 78 radio pulsars”. In: *Phys. Rev. D.* 76.4, 042001 (Aug. 2007), p. 042001. DOI: 10.1103/PhysRevD.76.042001. eprint: gr-qc/0702039.
- [230] B. P. Abbott et al. “Searches for Gravitational Waves from Known Pulsars with Science Run 5 LIGO Data”. In: *ApJ* 713 (Apr. 2010), pp. 671–685. DOI: 10.1088/0004-637X/713/1/671. arXiv: 0909.3583 [astro-ph.HE].
- [231] Josh Abadie et al. “Beating the spin-down limit on gravitational wave emission from the Vela pulsar”. In: *ApJ* 737.2 (2011), p. 93. DOI: 10.1088/0004-637X/737/2/93.
- [232] Vladimir Dergachev. personal communication. 2018.
- [233] *Initial LIGO strain noise data (PSDs)*. URL: https://labcit.ligo.caltech.edu/~jzweizig/distribution/LSC_Data/.
- [234] *Representative O1 noise PSDs*. H1 LIGO-G1600150, L1 LIGO-G1600151.
- [235] K. Wette et al. “Searching for gravitational waves from Cassiopeia A with LIGO”. In: *Class. Quant. Grav.* 25.23, 235011 (Dec. 2008), pp. 235011–+. DOI: 10.1088/0264-9381/25/23/235011. arXiv: 0802.3332 [gr-qc].

A. Bibliography

- [236] S. Bonazzola and E. Gourgoulhon. “Gravitational waves from pulsars: emission by the magnetic-field-induced distortion.” In: *A&A* 312 (Aug. 1996), pp. 675–690. eprint: [astro-ph/9602107](https://arxiv.org/abs/astro-ph/9602107).
- [237] Alexander H. Nitz et al. “1-OGC: The First Open Gravitational-wave Catalog of Binary Mergers from Analysis of Public Advanced LIGO Data”. In: *ApJ* 872, 195 (Feb. 2019), p. 195. DOI: 10.3847/1538-4357/ab0108. arXiv: 1811.01921 [gr-qc].
- [238] LIGO Scientific Collaboration and Virgo Collaboration. “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs”. In: *Phys. Rev. X* 9 (3 Sept. 2019), p. 031040. DOI: 10.1103/PhysRevX.9.031040. URL: <https://link.aps.org/doi/10.1103/PhysRevX.9.031040>.
- [239] V. Dergachev and M. Alessandra Papa. “First loosely coherent all-sky search for periodic gravitational waves in the O1 LIGO data”. In: *arXiv e-prints* (Feb. 2019). arXiv: 1902.05530 [gr-qc].
- [240] J. Ming et al. “Results from an Einstein@Home search for continuous gravitational waves from Cassiopeia A, Vela Jr. and G347.3”. In: *arXiv e-prints* (Mar. 2019). arXiv: 1903.09119 [gr-qc].
- [241] The LIGO Scientific Collaboration et al. “All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO O2 data”. In: *arXiv e-prints* (Mar. 2019). arXiv: 1903.01901 [astro-ph.HE].
- [242] David Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362 (2018), p. 1140. URL: <http://science.sciencemag.org/content/362/6419/1140>.
- [243] OpenAI. *OpenAI Five*. 2018. URL: <https://blog.openai.com/openai-five/>.
- [244] Oriol Vinyals et al. *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*. 2019. URL: [%7Bhttps://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/%7D](https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/).
- [245] S. Walsh et al. “Optimising the choice of analysis method for all-sky searches for continuous gravitational waves with Einstein@Home”. In: *arXiv e-prints* (Jan. 2019). arXiv: 1901.08998 [astro-ph.IM].
- [246] R. Prix. “Template-based searches for gravitational waves: efficient lattice covering of flat parameter spaces”. In: *Class. Quant. Grav.* 24 (July 2007), S481. eprint: 0707.0428.
- [247] K. Wette. “Lattice template placement for coherent all-sky searches for gravitational-wave pulsars”. In: *Phys. Rev. D*. 90 (Oct. 2014), p. 122010. arXiv: 1410.6882 [gr-qc].
- [248] Reinhard Prix. *Characterizing timing and memory-requirements of the F-statistic implementations in LALSuite*. Tech. rep. 2017. eprint: T1600531-v4+. URL: <https://dcc.ligo.org/LIGO-T1600531/public>.
- [249] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from [tensorflow.org](https://www.tensorflow.org/). 2015. URL: <https://www.tensorflow.org/>.
- [250] LIGO Scientific Collaboration. *LALSuite: FreeSoftware (GPL) tools for data-analysis*. URL: <https://www.lsc-group.phys.uwm.edu/daswg/projects/lalsuite.html>.
- [251] The LIGO Scientific Collaboration et al. “GW190425: Observation of a Compact Binary Coalescence with Total Mass $\sim 3.4M_{\odot}$ ”. In: *arXiv e-prints*, arXiv:2001.01761 (Jan. 2020), arXiv:2001.01761. arXiv: 2001.01761 [astro-ph.HE].

- [252] The LIGO Scientific Collaboration et al. “All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO O2 data”. In: *Phys. Rev. D* 100 (2 July 2019), p. 024004. DOI: 10.1103/PhysRevD.100.024004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.024004>.
- [253] J. Ming et al. “Results from an Einstein@Home search for continuous gravitational waves from Cassiopeia A, Vela Jr., and G347.3”. In: *Phys. Rev. D* 100 (2 July 2019), p. 024063. DOI: 10.1103/PhysRevD.100.024063. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.024063>.
- [254] Vladimir Dergachev and Maria Alessandra Papa. “Results from an extended Falcon all-sky survey for continuous gravitational waves”. In: *Phys. Rev. D* 101 (2 Jan. 2020), p. 022001. DOI: 10.1103/PhysRevD.101.022001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.101.022001>.
- [255] Karl Wette. “SWIGLAL: Python and Octave interfaces to the LALSuite gravitational-wave data analysis libraries”. In: *Submitted* ().
- [256] Yoshua Bengio et al. “Curriculum Learning”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 41–48. ISBN: 9781605585161. DOI: 10.1145/1553374.1553380. URL: <https://doi.org/10.1145/1553374.1553380>.
- [257] H. Shen et al. “Denoising Gravitational Waves with Enhanced Deep Recurrent Denoising Auto-encoders”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 3237–3241. URL: <https://ieeexplore.ieee.org/document/8683061>.

B. List of Tables

6.1.	All-sky searches: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1). The columns labeled f and \dot{f} give the frequency and spindown ranges covered by each search. Sensitivity depths in <i>italics</i> refer to 90%-confidence upper limits, while normal font refers to 95%-confidence. See appendix 6.8.2 for further details on the individual results.	72
6.2.	Directed and narrow-band searches: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1). The column labeled f gives the frequency range covered by each search (omitting \dot{f} and \ddot{f} search ranges). Sensitivity depths in <i>italics</i> refer to 90%-confidence upper limits, while normal font refers to 95%-confidence. See appendix 6.8.3 for further details on the individual results.	74
6.3.	S6-NineYoung- \mathcal{F} search: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1) for nine young supernova remnants [184]. All sensitivity depths refer to 95%-confidence. See appendix 6.8.3 for further details.	75
6.4.	Binary searches: measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (median and standard deviation, see Sec. 6.6.1). All sensitivity depths refer to 95%-confidence. See appendix 6.8.4 for further details on the individual results.	75
6.5.	Targeted searches for known pulsars: estimated \mathcal{D}_{est} and measured sensitivity depth $\mathcal{D}_{\text{meas}}$ (with respectively, median and standard deviation, see Sec. 6.6.1). All sensitivity depths refer to 95%-confidence. See appendix 6.8.5 for further details on the individual results.	77
7.1.	Definition of the two benchmark searches.	95
7.2.	WEAVE parameters and characteristics for the two searches.	96
7.3.	WEAVE characteristics for the ten test cases, each covering a frequency “slice” of $\Delta f = 50$ mHz, starting at f_0 , of the full searches defined in Table 7.1. The detection thresholds \mathcal{F}_{th} correspond to a false-alarm level of $p_{\text{fa}} = 1\%$ over the band Δf . $\mathcal{N}_{\Delta f}$ is the number of templates used per respective frequency band. $\text{CPU}_{\Delta f}$ denotes the search time in seconds for the respective Δf band on a single CPU core.	96
7.4.	Measured WEAVE “upper limit” sensitivity $\mathcal{D}^{90\%}$ at false-alarm level of $p_{\text{fa}} = 1\%$	97
7.5.	Detection probabilities in % of the best networks for each case at false alarm level $p_{\text{fa}} = 1\%$ and 90 % matched-filtering depth.	101
7.6.	Sensitivity depths $\mathcal{D}_{\text{DNN}}^{90\%}$ at false-alarm level of $p_{\text{fa}} = 1\%$ achieved by the network for the ten test cases. The respective matched filter depths can be found in Table 7.4.	102
7.7.	DNN computing cost (in seconds) for training, search and follow-up (using matched-filtering). The respective matched-filtering cost can be found in Table 7.2	103
8.1.	Definition of all-sky (two-detector) benchmark searches.	109
8.2.	Definition of directed benchmark searches, modeled after [253].	109
8.3.	WEAVE coherent matched-filtering search parameters and characteristics.	109

B. List of Tables

8.4. Sensitivity depths $\mathcal{D}_{\text{MF}}^{90\%}$ achieved by the WEAVE coherent matched-filtering search for the (two-detector) all-sky (a-s) and directed cases defined in Tables. 8.1 and 8.2. The all-sky sensitivity is improved by a factor of approximately $\sqrt{2}$ compared to the single-detector values reported in [30], as expected for coherent matched filtering. As training the CasA case $T = 10^6$ s, $f_0 = 1500$ Hz, “H1L1” required more GPU memory than available to us, we reduced the maximum frequency in the search to 1000 Hz. 110

8.5. Network sensitivity depths $\mathcal{D}_{\text{DNN}}^{90\%}$ for the (two-detector) all-sky (a-s) and directed search cases. The corresponding matched-filtering sensitivity depths are given in Table 8.4. As training the CasA case $T = 10^6$ s, $f_0 = 1500$ Hz, “H1L1” required more GPU memory than available to us, we reduced the maximum frequency in the search to 1000 Hz. 112

8.6. Network detection probabilities $p_{\text{det}}^{\text{DNN}}$ with 95 % error region for the (two-detector) all-sky (a-s) cases and directed cases for signals at the matched-filtering sensitivity depths $\mathcal{D}_{\text{MF}}^{90\%}$ given in Table 8.4. As training the CasA case $T = 10^6$ s, $f_0 = 1500$ Hz, “H1L1” required more GPU memory than available to us, we reduced the maximum frequency in the search to 1000 Hz. 113

C. List of Figures

3.1.	A circle of free-falling particles in the x-y-plane (a) is distorted (b) by a + polarized and (c) by a \times -polarized gravitational wave passing the setup in z- direction. Over time the effect periodically transitions from the top distortion via the circle to the bottom distortion and back.	9
3.2.	Sketch of the configuration of a laser-interferometric detector like LIGO or Virgo: a Michelson interferometer, where all mirrors and the beam splitter are suspended as isolation against ground motion. Additionally, mirrors in the arms are used to form Fabry-Pérot cavities in order to increase the effective arm length. The power recycling mirror forms a cavity with the entire interferometer. It reflects the light which normally returns to the laser back into the interferometer increasing the power (Credit: Fig. 9.3 in [39]).	10
3.3.	O2 noise budget of the LIGO Livingston detector (L1). The measured equivalent noise spectrum (blue) and the main explainable contributions: Technical, quan- tum, seismic and thermal noise. The narrow peaks are called lines. Some of them have known causes, e.g. 60 Hz is the power grid frequency in the US, but many also have unknown causes (Credit: Fig. 3 of [52]).	11
3.4.	Characteristic strain of the advanced LIGO detectors (aLIGO) at design sensitiv- ity and for the planned detectors Einstein Telescope (ET), Cosmic Explorer (CE) as well as the planned space-missions LISA and DECIGO. The color trapezoid are an estimate of the characteristic strain of several sources (cf. sec. 3.4). This figure was created with the GW plotter web application (http://www.gwplotter.com/ , which is based on [55]).	12
3.5.	GW150914: The first binary black hole merger observed by LIGO as velocity, separation and strain over time. The different phases of the coalescence, inspiral, merger and ringdown are illustrated by numerical relativity models of the black hole horizons at the top. The strain is reconstructed from the measured signal and compared to a numerical relativity calculation (Credit: Fig. 2 of [1]).	13
3.6.	Schematic of the structural layers of a neutron star: The neutron star is sur- rounded by a thin atmosphere of light gases. The outer crust consists of white- dwarf matter, heavy nuclei and electrons, the inner crust is made of nuclei, neu- trons and electrons. The core composition is not well known but in the outer regions it likely consists of neutrons, protons, electrons and muons [78].	15
3.7.	Gravitational-wave strain predictions for known pulsars. The grey curve is the strain sensitivity during the S5 science run of initial LIGO assuming the analysis of two years of data. The black curve and the dashed black curve are the projected sensitivity for CW searches with advanced LIGO and with the planned Einstein Telescope, respectively. The differently colored points represent strain predictions from different models for neutron star matter and magnetic fields. For more details and the original figure see [85].	16
4.1.	Geometric relationship between SSB, detector and source for an incoming gravi- tational wave. The reference frames along the line of sight \hat{n} are the (\hat{l}, \hat{m}) frame usually aligned with source parameters and the blue (\hat{i}, \hat{j}) frame aligned with Earth rotation axis. The inclination angle of the source relative to the line of sight is denoted as ι	20

C. List of Figures

5.1. Machine-learning algorithms find transformations to represent data more simply. In the coordinate system (Cartesian) in (a) the points are clearly not separable along one variable. If we apply a transformation to polar coordinates (b), we can separate the points along one variable.	32
5.2. A dense neural network with three inputs (green), two outputs (red) and three hidden layers (blue).	33
5.3. A single neuron of a neural network. In this example the neuron has two inputs, two weights \vec{w} , a bias b and an activation function a	33
5.4. Six important activation functions in neural networks: Sigmoid, rectified linear unit (ReLU), Heaviside step function, leaky ReLU, exponential linear unit (ELU) and scaled ELU (SELU).	34
5.5. The training loop of a neural network consists of the forward pass and the backward pass. In the forward pass the network is evaluated for the inputs from the training set. The output is a prediction of the label of this input sample. The predictions are then compared to the labels of the training set to calculate the loss. In the backward pass the gradient of the loss with respect to the weights is calculated and then used to update the weights of the network via <i>gradient descent</i>	36
5.6. Detection probability vs. training time of a noise-signal classifier for CWs: In (a) the network clearly overfits the data. The training performance reaches the desired performance marked in red and the validation set performance remains low although both sets contain samples of the same distribution. In (b) the network underfits the data. Training and validation performance are very similar but far from the desired performance marked by the red line.	42
5.7. Fitting a fifth degree polynomial (blue, dotted) from some noisy data points (blue) with a cubic polynomial (red), with a 20th degree polynomial (orange) and with a fifth degree polynomial (green). Only the fifth degree polynomial fit approximates the original function with a low error and would give good predictions for newly added data points. The 20th degree polynomial overfits and the third degree polynomial underfits the data.	43
5.8. Application of a size 3 filter in a convolutional layer: In a simple convolutional layer (a) the filter is moved over the input a , giving a smaller output b . Often the edges are zero-padded (b) to keep the output the same size as the input. If the input has multiple channels (two channels in example (c)) the filter has three weights per channel and the receptive field extends over all channels. If multiple convolutional layers are stacked (e.g. two filter size 3 layers in (d)), the network's receptive field, measured on the input a , grows.	46
5.9. (a) Residual connections (red) allow the network to skip for example the layers $l + 1$ or $l + 2$. The gradient calculation is therefore less affected by the weights of the circumvented layers improving the gradient calculation for earlier layers. Inception modules (b) use parallel layers with different filter sizes to allow more diverse features to be detected. In this figure the notation Conv1D(c, f) denotes a 1D convolutional layer with a filter size of f , c feature maps and a stride of 1. MaxPooling(3) is a max pooling layer with a receptive field of 3 and a stride of 3. In the end the columns in the Inception module are combined by concatenation resulting in 96 channels. The previous layer had 64 channels.	49
5.10. Inception-ResNet-A block as used in Ch. 8 with the same notation as in Fig. 5.9. The residual connection (red) skips the entire Inception module. Compared to the original block (see Fig. 16 in [175]) the 3×3 filters were replaced by one dimensional filters of size 5. The other InceptionResNet-v2 blocks were adapted similarly.	49

6.1.	Detection probability p_{det} versus sensitivity depth \mathcal{D} for the S6-CasA-StackSlide-\mathcal{F} search (cf. Table 6.2 and Sec. 6.8.3), using a detection threshold of $2\overline{\mathcal{F}}_{\text{th}} = 8$. The squares indicate the results from a simulation in Gaussian noise, while the solid line gives the best-fit sigmoid of Eq. (6.54).	65
6.2.	Histogram of recovered loudest $2\overline{\mathcal{F}}$ values for repeated searches on signal injections at fixed sensitivity depth $\mathcal{D} = 86 \text{ Hz}^{-1/2}$ (with all other signal parameters randomized), using the search setup of the S6-CasA-StackSlide-\mathcal{F} directed search. The vertical line indicates the resulting threshold value $2\overline{\mathcal{F}}_{\text{th}} = 7.995$ corresponding to $p_{\text{det}} = 90\%$ for this injection set.	65
6.3.	Sensitivity depth versus detection threshold. Boxes and solid lines indicate the piecewise-linear interpolation through the obtained thresholds at different depths of an injection-recovery simulation, using the S6-CasA-StackSlide-\mathcal{F} search setup ([185] and Sec. 6.8.3).	66
6.4.	Comparison of estimated and simulated sensitivity depth $\mathcal{D}^{90\%}$ as a function of threshold $2\overline{\mathcal{F}}_{\text{th}}$ for the S6-AllSky-StackSlide-\mathcal{F} search [207]. The solid line shows the UL estimate for the <i>box search</i> , and the squares (\square) show the corresponding simulated ULs. The dashed line indicates the estimate for the <i>zero-mismatch case</i> , and the crosses (\times) are for the simulated zero-mismatch ULs. In the box search we observe an increasing divergence at decreasing thresholds due to noise effects, discussed in Sec. 6.5.1.	67
6.5.	Distribution of the loudest $2\overline{\mathcal{F}}$ for a box search on pure Gaussian noise, using the S6-AllSky-StackSlide-\mathcal{F} search setup.	68
6.6.	Loudest $2\overline{\mathcal{F}}$ distribution for a box-search (using the S6-AllSky-StackSlide-\mathcal{F} setup) with signals at a depth of $\mathcal{D} = 20 \text{ Hz}^{-1/2}$ (left plot) and $\mathcal{D} = 46 \text{ Hz}^{-1/2}$ (right plot). The black histogram shows the assumed distribution for sensitivity estimation in Eq. (6.37), and the lighter color shows the histogram obtained in a Monte-Carlo <i>simulation</i> with signals injected in Gaussian noise.	69
6.7.	Comparison of estimated and simulated sensitivity depth $\mathcal{D}^{90\%}$ as a function of the threshold $2\overline{\mathcal{F}}_{\text{th}}$ for two targets of the multi-directed search setup 01-MD-StackSlide-\mathcal{F} . The solid lines show the UL estimate for a <i>box search</i> , while the squares (\square) show the corresponding simulated ULs. The dashed lines indicate the estimate for the <i>zero-mismatch case</i> , and the crosses (\times) are for the simulated zero-mismatch ULs. The upper group of curves are for the target Vela Jr., while the lower group of curves are for Cas A.	70
6.8.	Estimated ($-$) and simulated (\square) sensitivity depth versus threshold $2\overline{\mathcal{F}}_{\text{th}}$ for the S6-AllSky-StackSlide-\mathcal{F} (+FUP) search setup, illustrating the effect of the template-maximization in the estimate (discussed in Sec. 6.5.1). The triangles (Δ) and dashed lines show the measured upper-limit depth $\mathcal{D}_{\text{meas}}^{\text{med}}$ in the initial S6-AllSky-StackSlide-\mathcal{F} search [207], and the diamond (\diamond) shows the corresponding result from the follow-up (FUP) search [201] (threshold $2\overline{\mathcal{F}}_{\text{th}} = 6.1$).	71
6.9.	Estimated ($-$) and simulated (\square) sensitivity depth versus threshold $2\overline{\mathcal{F}}_{\text{th}}$ for the S6-CasA-StackSlide-\mathcal{F} search setup [185]. The published upper limits are plotted as triangles (Δ), while the diamonds (\diamond) show the simulated depths if we incorporate the bug found in the original UL procedure.	73
6.10.	<i>Stair-case plot</i> : histogram (over 10^6 repeated trials) of $2\mathcal{F}_2^* = \max_k \tilde{x}_k ^2$ for Fourier transforms of Gaussian-noise timeseries, using different oversampling factors (a)–(d), where oversampling = 1 corresponds to the original FFT frequency resolution. <i>Solid thin line</i> : corresponding best-fit theoretical model Eq. (6.73) with an effective number of templates \mathcal{N}'	92
7.1.	Illustration of the general network architecture used in this study.	97

C. List of Figures

7.2. Validation detection probability for $T = 10^5$ s, $f_0 = 1000$ Hz, “H1” for training with training sets containing 10, 100, 1000, 10000 and 50000 signals. 98

7.3. Validation detection probability $p_{\text{det}}^{\text{DNN}}$ of the DNN versus training time (or mean trained epoch) for 100 different network instances trained for each of four test cases: (a) $T = 10^5$ s, $f_0 = 20$ Hz, “H1”, (b) $T = 10^5$ s, $f_0 = 1000$ Hz, “H1”, (c) $T = 10^6$ s, $f_0 = 20$ Hz, “H1” and (d) $T = 10^6$ s, $f_0 = 1000$ Hz, “H1”, all trained on Nvidia GTX 750. The solid horizontal line denotes the matched-filtering detection performance of $p_{\text{det}} = 90\%$ 99

7.4. Validation detection probability $p_{\text{det}}^{\text{DNN}}$ of the DNN versus training time for a single network trained on Nvidia TITAN V for the case $T = 10^6$ s, $f_0 = 1000$ Hz, “H1”. 100

7.5. ROC-curve: Detection probability p_{det} versus p_{fa} for the 10^5 s search (left) and the 10^6 s search (right). The solid red lines indicate the measured ROC curves for matched filtering. 101

7.6. Detection probability p_{det} versus injection frequency f for networks trained at five different frequencies and for a network trained with signals drawn from the full frequency range (solid black line). The dashed vertical lines mark the respective training frequencies for the five “specialized” networks. The horizontal dashed line represents the coherent matched filtering detection performance. 102

7.7. Detection probability p_{det} versus injected spin-down \dot{f} for networks trained at give different frequencies. The green shade in the middle marks the 10^{-10} Hz/s wide spin-down band the networks were trained on. The x-axis is plotted as a symmetric logarithm, i.e. logarithmical for the larger negative values, linear for $|\dot{f}| < -10^{-10}$ Hz/s and logarithmical for the larger positive values. The red shades at the edges illustrates where we start losing SNR purely by the network input window being smaller than the widest signals. 103

7.8. Detection probability p_{det} versus injection depth \mathcal{D} for networks trained on the respective matched-filtering depth $\mathcal{D}^{90\%}$ (indicated by the vertical red line with the at 90%). The blue vertical line gives the sensitivity depth for the DNN at 90% detection probability. 104

8.1. Detection probability p_{det} versus injection frequency f for the all-sky networks trained at five different frequencies and for a network trained with signals drawn from the full frequency range (solid black line). The dashed vertical lines mark the respective training frequencies for the five “specialized” networks. The solid red horizontal line represents the coherent matched-filtering detection performance. The shaded areas around each curve show the 95% error regions. The analogous single-detector result is found in Fig.6a of [30]. 114

8.2. Detection probability p_{det} versus injection frequency f for networks trained at three different frequencies for the CasA and the G347 target, respectively, and for a network trained with signals drawn from the full frequency range (solid black line). The dashed vertical lines mark the respective training frequencies for the three networks. The solid red horizontal line represents the coherent matched-filtering detection performance. The shaded areas around each curve show the 95% error regions. 114

8.3. Detection probability p_{det} versus injection depth \mathcal{D} for networks trained on the respective matched-filtering depth $\mathcal{D}_{\text{MF}}^{90\%}$ (indicated by the vertical solid line with the diamond at 90%). The second vertical line which crosses the DNN curve at 90% gives the sensitivity depth for the DNN at 90% detection probability. The shaded region around the DNN curve is the 95% error region. The respective errors for the matched-filtering results are smaller than the thickness of the curve. 115

8.4.	Detection probability p_{det} versus injected spindown \dot{f} for networks trained at different frequencies. The x-axis is plotted as a symmetric logarithm, i.e. logarithmic for the larger negative values, linear for $ \dot{f} < -10^{-10}$ Hz/s and logarithmic for the larger positive values. The vertical dashed lines mark the minimal spindown \dot{f} used in the training set. Its absolute value increases with frequency. The maximal used spindown for all cases is 0 Hz/s (dotted line). The shaded areas around each curve show the 95 % error regions.	116
8.5.	Detection probability p_{det} versus injected second order spindown \ddot{f} for networks trained at different frequencies. All other parameters were the same as during training. The x-axis is plotted as a symmetric logarithm, i.e. logarithmic for the larger negative values, linear for $ \ddot{f} < -10^{-20}$ Hz/s and logarithmic for the larger positive values. The vertical dashed lines mark the maximal second order spindown \ddot{f} used in the training set, which increases with frequency. The minimal used second order spindown for all cases is 0 Hz/s ² (dotted line). The shaded areas around each curve show the 95 % error regions.	117
8.6.	Detection probability p_{det} as a function of the sky-position of injected signals in equatorial coordinates (Hammer projection). The detection probability is measured at fixed SNR $\rho = 8.94$. In (a)-(d) the respective sky position of CasA or G347 is marked by a white plus.	118
8.7.	Duty factor vs detection probability of an all-sky DNN in Gaussian and real noise. The solid red curve with its shaded region represents the behaviour of matched filtering on Gaussian noise, the blue crosses represent the DNN's performance on Gaussian noise and the yellow circles represent the DNN's performance on real LIGO O1 detector noise. The error bars indicate the 95 % confidence interval.	119
8.8.	Histogram of the distribution of DNN detection statistic values (predictions) for 1000 real-noise input samples. The two distributions correspond to two different start-times, with similar duty factors $\sim 82\%$ and for the same 5 Hz band around 200 Hz. In one case (blue) a disturbance in the data results in a long tail of higher statistic values, which leads to a higher detection threshold at fixed false-alarm, thereby reducing detection probability compared to the undisturbed case (orange).120	

D. List of Publications

1. Christoph Dreissigacker, Reinhard Prix, and Karl Wette. “Fast and accurate sensitivity estimation for continuous-gravitational-wave searches”. In: *Phys. Rev. D* 98 (8 Oct. 2018), p. 084058. DOI: 10.1103/PhysRevD.98.084058. URL: <https://link.aps.org/doi/10.1103/PhysRevD.98.084058>
2. Karl Wette et al. “OctApps: a library of Octave functions for continuous gravitational-wave data analysis”. In: *Journal of Open Source Software* 3.26 (2018), p. 707. DOI: 10.21105/joss.00707
3. Christoph Dreissigacker et al. “Deep-learning continuous gravitational waves”. In: *Phys. Rev. D* 100.4, 044009 (Aug. 2019), p. 044009. DOI: 10.1103/PhysRevD.100.044009. arXiv: 1904.13291 [gr-qc]
4. Christoph Dreissigacker and Reinhard Prix. “Deep-learning continuous gravitational waves: Multiple detectors and realistic noise”. In: *Phys. Rev. D* 102 (2 July 2020), p. 022005. DOI: 10.1103/PhysRevD.102.022005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.022005>

From 2016 to 2018 the author was member of the LIGO Scientific Collaboration. Publications co-authored as part of the LIGO collaboration are not listed here.

E. Curriculum Vitae

Christoph Dreißigacker

Date of birth 18th February 1990
Place of birth Hildesheim, Germany
Email christoph.dreissigacker@aei.mpg.de

Education and scientific career

2000 - 2009 Abitur - Gymnasium Marienschule Hildesheim

2010 - 2014 BSc Physik - Leibniz Universität Hannover
Bachelor thesis Numerische Studien der Schrödinger-Newton Gleichung
Numerical studies of the Schrödinger-Newton equation
Supervisor: Manfred Lein

2013 - 2016 MSc Physik - Leibniz Universität Hannover
Master thesis Tidal deformations of black holes in non axially symmetric configurations
Supervisor: Badri Krishnan, Domenico Giulini

2016 - 2020 Dr. rer. nat. - Albert-Einstein-Institute, Leibniz Universität Hannover
Thesis title Searches for Continuous Gravitational Waves: Sensitivity Estimation and Deep Learning as a Novel Search Method
Doctoral advisor: Reinhard Prix

2016 - 2018 Member of the LIGO Scientific Collaboration

2020 - Present Postdoctoral researcher - Albert-Einstein-Institute

Employment history

Sep 2009 - St. Bernward Krankenhaus, Hildesheim

Mai 2010 *Compulsory civil service - Zivildienst*

Oct 2011 - Leibniz Universität Hannover

Mar 2016 *Student assistant*
Tutoring students in the basics of theoretical physics

Oct 2016 - Leibniz Universität Hannover

Oct 2017 *Head tutor for Karsten Danzmann*
Co-managing tuition for students of the lectures “Gravitational Physics” and “Mechanics and Relativity”
Selection and supervision of tutors

Languages

- German : native
- English : fluent
- French : basic

IT Skills

- Python
- Octave
- Keras (Tensorflow)

F. Acknowledgements

During the years of researching and writing this thesis, many persons helped me and my development as a scientist. Without them this thesis would not have been possible. First of all I want to thank Reinhard Prix. He proposed many of the ideas for my research and introduced me to the field of CW data analysis. His comments and reviews certainly improved my results significantly.

A special thanks goes also to Bruce Allen and Maria Alessandra Papa for letting me join the great Data analysis and CW groups of the AEI. I am also grateful for all the helpful and fun interactions with all the members of the AEI CW and data analysis group and with my great office mates.

I also want to thank all my co-authors on the published papers: Reinhard Prix, Karl Wette, Rahul Sharma, Ruining Zhao and Chris Messenger. Additional thanks to all who agreed to be referees and defense committee members for my thesis: Maria Alessandra Papa, Chris Messenger, Alicia Sintes, Reinhard Prix, Domenico Giulini and Benno Willke. I also want to thank the referees for the comments which I addressed to create this final version of the thesis.

I am also thankful to Henning Fehrmann, Carsten Aulbert and Alexander Post for all the help related to the ATLAS computing cluster. I also want to thank Karsten Danzmann for giving me exciting teaching opportunities and Jonas Junker for the joint teaching work.

Another thanks goes to the people who were reading through chapters of this thesis: Reinhard Prix, Marlin Schäfer and Julian Westerweck. And a very special thanks goes to Nina and my parents for the proof-reading and the great support in all this time, especially in the final weeks of writing this thesis under special home-office circumstances.