
Applications of Machine Learning to Gravitational Waves

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)



vorgelegt dem Rat der
PHYSIKALISCH-ASTRONOMISCHEN FAKULTÄT
der
FRIEDRICH-SCHILLER-UNIVERSITÄT JENA
von

Mgr. Ondřej ZELENKA

geboren am 14. Februar 1995 in Most, Tschechische Republik

Gutachter:

1. Prof. Dr. Bernd Brügmann

Theoretisch-Physikalisches Institut, Friedrich-Schiller-Universität Jena

2. Dr. hab. Michał Bejger

Nicolaus Copernicus Astronomical Center of the Polish Academy of Sciences

3. Prof. Dr. Alicia Magdalena Sintés Olives

Departament de Física & IAC3, Universitat de les Illes Balears

Tag der Disputation: 2. 11. 2023

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA

Abstract

PHYSIKALISCH-ASTRONOMISCHE FAKULTÄT

Theoretisch-Physikalisches Institut

zur Erlangung des akademischen Grades doctor rerum naturalium (Dr. rer. nat.)

Applications of Machine Learning to Gravitational Waves

by Mgr. Ondřej ZELENKA

Gravitational waves, predicted by Albert Einstein in 1916 and first directly observed in 2015, are a powerful window into the universe, and its past. Currently, multiple detectors around the globe are in operation. While the technology has matured to a point where detections are common, there are still unsolved problems. Traditional search algorithms are only optimal under assumptions which do not hold in contemporary detectors. In addition, high data rates and latency requirements can be challenging.

In this thesis, we use new methods based on recent advancements in machine learning to tackle these issues. We develop search algorithms competitive with conventional methods in a realistic setting. In doing so, we cover a mock data challenge which we have organized, and which served as a framework to obtain some of these results. Finally, we demonstrate the power of our search algorithms by applying them to data from the second half of LIGO's third observing run. We find that the events targeted by our searches are identified reliably.

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA

Zusammenfassung

PHYSIKALISCH-ASTRONOMISCHE FAKULTÄT
Theoretisch-Physikalisches Institut

zur Erlangung des akademischen Grades doctor rerum naturalium (Dr. rer. nat.)

Anwendungen des maschinellen Lernens auf Gravitationswellen

von Mgr. Ondřej ZELENKA

Gravitationswellen, die 1916 von Albert Einstein vorhergesagt und 2015 erstmals direkt beobachtet wurden, sind ein wichtiges Fenster zum Universum und seiner Vergangenheit. Derzeit sind mehrere Detektoren rund um den Globus in Betrieb. Auch wenn die Technologie inzwischen so weit ausgereift ist, dass Detektionen häufig vorkommen, gibt es immer noch ungelöste Probleme. Traditionelle Suchalgorithmen sind nur unter Annahmen optimal, die bei modernen Detektoren nicht zutreffen. Darüber hinaus können hohe Datenraten und Latenzzeiten eine Herausforderung darstellen.

In dieser Arbeit verwenden wir neue Methoden, die auf den jüngsten Fortschritten im Bereich des maschinellen Lernens basieren, um diese Probleme anzugehen. Wir entwickeln Suchalgorithmen, die in einem realistischen Umfeld mit konventionellen Methoden konkurrieren können. Dabei behandeln wir eine von uns organisierte Mock Data Challenge, eine kompetitive Untersuchung verschiedener Methoden basierend auf realistischen, jedoch künstlich generierten Daten, welche als Rahmen zur Erlangung dieser Ergebnisse diente. Schließlich demonstrieren wir die Leistungsfähigkeit unserer Suchalgorithmen, indem wir sie auf die Daten der zweiten Hälfte des dritten LIGO-Beobachtungslaufs anwenden. Wir stellen fest, dass die Ereignisse, auf die unsere Suchalgorithmen abzielen, zuverlässig wiedergefunden werden.

Notation and abbreviations

New terms are first mentioned in *italics*, and new variables are defined using $:=$. We make frequent use of the *solar mass* $M_\odot \doteq 1.99 \cdot 10^{30}$ kg. Inferred parameters are given as, e.g., $m_1 = 8.2_{-1.6}^{+1.4} M_\odot$, where the first value refers to the median and the additive sub- and superscript refer to the 10th and 90th percentile, respectively.

Throughout the text, matrix notation is used. Bold uppercase symbols refer to matrices, e.g. $\mathbf{W} = (W_{ij})_{i=1,j=1}^{m,n}$, while bold lowercase symbols are vectors: $\mathbf{a} = (a_i)_{i=1}^n$, $\mathbf{b} = (b_j)_{j=1}^m$. Derivatives of scalar functions with respect to vectors and matrices are taken element-wise:

$$\frac{\partial \mathcal{C}}{\partial \mathbf{a}} := \left(\frac{\partial \mathcal{C}}{\partial a_i} \right)_{i=1}^n, \quad \frac{\partial \mathcal{C}}{\partial \mathbf{W}} := \left(\frac{\partial \mathcal{C}}{\partial W_{ij}} \right)_{i=1,j=1}^{m,n}. \quad (1)$$

Matrix-vector multiplication is defined as

$$\mathbf{W}\mathbf{a} := \left(\sum_{j=1}^n W_{ij} a_j \right)_{i=1}^m, \quad \mathbf{b}^T \mathbf{W} := \left(\sum_{i=1}^m b_i W_{ij} \right)_{j=1}^n. \quad (2)$$

The Hadamard (element-wise) product \odot and the tensor product \otimes are defined as

$$\mathbf{a} \odot \mathbf{b} := (a_i b_i)_{i=1}^n \quad (m = n), \quad \mathbf{a} \otimes \mathbf{b} := (a_i b_j)_{i=1,j=1}^{n,m}. \quad (3)$$

Finally, we define the cross-correlation operator over discrete samples

$$\mathbf{a} \star \mathbf{b} := \left(\sum_{i=1}^n a_i b_{i+j-1} \right)_{j=1}^{m-n+1}. \quad (4)$$

The following abbreviations are used, and are introduced at their first appearance:

ASD amplitude spectral density.

BBH binary black hole.

BCE binary cross entropy.

BH black hole.

BHNS black hole-neutron star.

BNS binary neutron star.

CBC compact binary coalescence.

CNN convolutional neural network.

DAIN deep adaptive input normalization.

DNN deep neural network.

EOB effective-one-body.

FAP false alarm probability.

FAR false alarm rate.

GR general relativity.

GW gravitational wave.

ML machine learning.

MLGWSC-1 first machine learning gravitational-wave search challenge.

NR numerical relativity.

NS neutron star.

PSD power spectral density.

ROC receiver operating characteristic.

SGD stochastic gradient descent.

SNR signal-to-noise ratio.

USR unbounded softmax replacement.

Contents

Abstract	v
Zusammenfassung	vii
Notation and abbreviations	ix
1 Introduction	1
2 Gravitational waves	7
2.1 Basics of gravitational-wave theory	7
2.2 Compact binary coalescence	14
2.3 Interferometric gravitational wave detectors	18
2.4 Detector data analysis	23
3 Machine learning	27
3.1 Basics of machine learning	28
3.2 Training a model	30
3.3 Neural networks	37
3.4 Regularization	44
4 Single-detector searches and training strategies	47
4.1 Experimental setup	48
4.2 Unbounded Softmax replacement	51
4.3 Training strategies	54
4.4 Search evaluation	59
5 First machine learning gravitational-wave search challenge	63
5.1 Test data	63
5.2 Evaluation	67
5.3 Results	68
5.4 Submission by the TPI FSU Jena team	71

5.5	Correction of performance issues	79
5.6	Validation	84
5.7	Application to O3b data	86
6	Conclusion	93
A	Experiment with an extended mass range	99
B	Segment keys	103
	Bibliography	107
	Acknowledgements	153
	Ehrenwörtliche Erklärung	155

Chapter 1

Introduction

Few discoveries of modern physics have attracted as much attention as gravitational waves (GWs), first predicted by Albert Einstein as a solution of his gravitational field equations [1]. They are wave-like disturbances of the spacetime geometry, radiated by high-energy events and propagating through the universe at long distances.

The first experimental evidence of their existence has been provided by radio observations of the Hulse-Taylor binary pulsar. First discovered in 1974, measurements of its orbital period matched the prediction of its decay by GW emissions very accurately [2, 3]. This was later named the first indirect detection of GWs and awarded the 1993 Nobel Prize in Physics [4].

The first direct detection, however, had to wait until 14 September 2015, when GWs from a binary black hole (BBH) merger were observed by the Laser Interferometer Gravitational Observatory (LIGO) [5–8]. During the following years, further groundbreaking detections occurred, such as the first observation of a binary neutron star (BNS) merger [9, 10], or the observation of a BBH merger with an unprecedented total mass of 150 solar masses [11, 12]. In addition, the first BNS event was observed through both gravitational and electromagnetic waves, marking the first *multi-messenger observation* using GWs [13–16].

Black holes (BHs) of masses up to $\sim 100M_{\odot}$ are believed to be the final stage of massive stars. Supermassive BHs with masses $\gtrsim 10^5M_{\odot}$ have been discovered and confirmed through observations of the motion of nearby stars [17, 18] and accreting matter, and even direct electromagnetic observations through the Event Horizon Telescope [19, 20]. However, their formation mechanism is an open question. In the same spirit, the population of BHs in the mass range $[100M_{\odot}, 10^5M_{\odot}]$ is yet unknown [21]. GW observations of merging compact binaries are expected to shed light on these issues as well as the overall mass distribution of BHs in the universe. BH population and formation are critical questions of not only astronomy, but also cosmology [22–24].

A lower-mass star may leave, as its remnant, a neutron star (NS) instead, which consists of extremely dense neutron-degenerate matter. The behavior of this matter includes many effects which are difficult to reproduce in a laboratory. GW signals from BNS mergers contain imprints of these effects and their observations expand our understanding of extremely dense matter and, by extension, high-energy particle physics [25, 26]. Furthermore, some elements crucial for life on Earth are created mostly in BNS mergers [27–29].

BHs and NSs, along with white dwarves [30], are called *compact objects*. They are known to exist in binaries originating from multiple formation mechanisms [31–33]. These binaries gradually lose energy through GW radiation, until they eventually merge into a single compact object (typically a BH). The late inspiral and merger stages of compact binaries are some of the most extreme events in our universe, which makes them a laboratory for strong gravity, capable of revealing deviations from the general theory of relativity [34–36]. Other significant, although yet undetected, sources include core-collapse supernovae [37, 38] and rapidly rotating NSs, emitting continuous GWs through a multitude of effects [39–43]. For these reasons, GW observations are a highly effective window into our universe, its composition and history [44, 45].

As GW research gradually attracted attention, multiple experimental detection methods were conceived, such as interferometric detectors, resonant mass antennae [46–48] or pulsar timing arrays [44, 49]. Specifically, the interferometric detectors have been successful at detecting signals from mergers of compact binaries and are well suited to this problem.

At the same time, the noise characteristic of contemporary detectors is complicated, as the noise is colored and contains noise transients which may be mistaken for astrophysical signals [50]. Due to this as well as comparatively weak signals, analysis of detector data is a complex task, both searching for astrophysical origin signals buried in noise, and recovering source properties, such as component masses and sky location, and in the case of NSs, tidal deformabilities. In this work, we focus purely on the detection problem.

Modeled searches rely on template banks which densely cover the parameter space of potential sources. Each template is matched against the detector data and close matches are marked as candidate events. They are called matched-filtering searches and have been proven to be optimal in stationary Gaussian background noise [51]. Elaborate systems are employed to limit the effect of the deviations from the expected noise distribution. Complicated parameter spaces require large template banks to

ensure sufficient coverage.

The LIGO-Virgo collaboration employs three modeled searches: PyCBC [51–57], GstLAL [58–61], and MBTA [62, 63], as well as the minimally modeled search coherent WaveBurst (cWB) [64–68]. For a summary evaluation of GW catalogs [69–72], they are applied in their offline regime after the observing period has ended, to the final calibrated and cleaned dataset. To provide low-latency alerts for rapid electromagnetic follow-up and potential multi-messenger detections, all the pipelines above in their online configurations as well as the SPIIR [73–75] pipeline are used. Further pipelines are in place to infer properties of GW sources [6, 72, 76–80].

As detector upgrades reduce the noise power and increase detector sensitivity [81], certain effects become observable. These include highly eccentric orbits of a compact binary [82], precession due to strong misaligned spins of its components [83], or higher modes [84]. This adds new dimensions to the parameter space and grows the template bank by a large factor, which increases the computational times of a search. At the same time, multi-messenger astronomy requires very fast online searches and sky localization pipelines while allowing higher false alarm rates than offline analyses.

Furthermore, projects for new detectors are underway, namely the Einstein telescope [85–88] or Cosmic Explorer [89, 90]. They are expected to detect mergers of intermediate-mass BHs with both stellar mass BHs and other intermediate-mass BHs [91–93], providing a unique window into the BH populations in our universe, and even a GW stochastic background, produced by a variety of cosmological sources [94–102].

The Laser Interferometer Space Antenna (LISA) [103–105] mission is currently scheduled to launch in 2037. It is the project of a space-borne detector most sensitive at frequencies $\sim 10^{-2}$ Hz and it will allow observations of sources undetectable by Earth-based detectors most sensitive at ~ 100 Hz. Among the most significant of LISA sources are *extreme mass ratio inspirals*: inspirals of stellar mass compact objects into supermassive BHs. Observing such events will allow us to map the spacetime around a supermassive BH and contribute to testing general relativity to an unprecedented degree of accuracy, as well as results on BH uniqueness [44, 106]. Unlike fairly short transients detected by LIGO, these sources are expected to be in the sensitive band of LISA for up to months at a time and likely overlap.

Ideas to extend the network of GW detectors go as far as construction of gravitational observatories on the Moon [107, 108]. All of these projects, while opening up new possibilities, will also pose new challenges for data analysts of the future.

Machine learning (ML) is an interesting set of methods focused on algorithms which improve their performance as they repeat their assigned task. Since its conception in the 1950s [109], interest in the subject has wavered, and began to attract large numbers of researchers in the 1990s and 2000s. This new rise was due to advancements in hardware, software [110] and theory [111–113], as well as availability of data, which led to a rapid increase of the potential of ML. By now, its applications range from classification and prediction [114, 115], through generation of artificial data [116–118] to playing games, such as chess or go, at a level where they can beat the best human players [119, 120].

Deep neural networks (DNNs) are a class of ML models, made of large numbers of very simple building blocks. When built in a suitable manner, they can perform complex tasks efficiently despite their inherent simplicity [111, 121]. Convolutional neural networks (CNNs) are particularly well suited to image processing, as evidenced by the great success of the AlexNet [122] and a subsequent improvement using a very deep 152-layer network by a ML group from Microsoft [123], which were another driving moment in ML history. CNNs are also well suited to time series data, such as speech commands and their recognition by smartphones. Along with their development, multiple DNN frameworks have been built, such as PyTorch [124], TensorFlow [125, 126], Caffe [127], or Theano [128].

Let us also mention a currently very popular application of ML methods, a language model called ChatGPT [129]. “ChatGPT is a large language model developed by OpenAI, based on the GPT-3.5 architecture. It has been trained on a massive corpus of text data and is capable of generating human-like responses to a wide range of prompts. While ChatGPT is not specifically designed for GW data analysis, its natural language processing capabilities could potentially be applied to tasks such as text-based analysis of scientific papers or communication between researchers in the field.”¹

As such, this powerful method has found applications in many fields of physics [130–133]. Most notable examples are in high-energy physics [134–145], astronomy [146–155] or cosmology [156–158]. In connection to applications in signal processing [159–161], this includes GW astronomy. The first steps in this direction have been done by [162, 163], which successfully trained CNNs to recognize BBH signals in isolated one-second samples of data in a single detector to a similar degree of accuracy as traditional classification methods. These results sparked a large influx of new ML

¹This description was generated by the ChatGPT chatbot as well as the citation [129] in the \LaTeX format, except for the URL, which was incorrect in the generated citation.

applications to search for GWs from compact binaries, including but not limited to CNNs [164–176], as well as for other sources [177–188].

At the same time, applications to other GW data analysis problems also appeared, such as parameter estimation [189–193], denoising [194–196], or fast waveform generation [197–200], required to accelerate matched-filter based analyses [201, 202]. Other ML-based results include seismic noise removal [203–206] and advanced instrument control techniques [207], or even more loosely related projects such as BH population inference [208]. For a summary, see [209, 210].

Many of these applications are also under investigation by the G2net network [211], which I have joined and participated in its activities. It has organized two mock data challenges in GW detection using ML-based methods on the Kaggle platform [212–214].

In this thesis, we follow up the pioneering papers [162, 163]. While important results in their own frameworks, several crucial steps were yet to be done to become a competitive search [164]. Primarily, they only used isolated samples in which the waveforms were aligned in a small window, and the algorithm would merely assign a ranking statistic to each sample, rather than taking a longer stream of data and searching for events. Furthermore, the false alarm probabilities were much higher than acceptable in conventional analyses [215], and they were in disagreement about the effect of the loudness of signals presented to the network during different stages of training.

These are the issues on which we focus in the projects described in this thesis: first, we extend a search very similar to that of [163] to a lower false alarm rate and into a full search algorithm. We apply it to a month-long continuous dataset and show that its performance is close to an equivalent matched-filter search at false alarm rates down to 1 per month. In the process, we develop a solution to the efficiency collapse due to rounding errors in the Softmax layer, and optimize the training procedure to achieve reliable convergence.

In the following project, we develop the search further into a two-detector search capable of distinguishing astrophysical signals from noise transients, and we find that our searches perform close to established matched-filter searches. This is done in the framework of the mock data challenge which we have organized [216]. Finally, we apply our searches to the latest data from the LIGO detectors [72]. We find that they reliably reject noise transients and recover the events our searches are trained to identify at false alarm rates < 5 per month, and signals coming from sources not

covered by our training procedure provided they are loud enough.

The organization of this thesis is as follows: In chapter 2, basics of GWs, their radiation from binary BHs, and their detection are covered. In chapter 3, foundations of ML methods are introduced, and Chapter 4 presents results of a project on using ML methods to search for BBH signals in strain data from a single GW detector. Chapter 5 presents its extension to two detectors as part of a mock data challenge on this topic, the MLGWSC-1, further developments of the search algorithm, and an application to O3b data.

Chapter 2

Gravitational waves

Geometric units $G = c = 1$ are employed, unless specified otherwise. Occasionally we write c explicitly for clarity. For full spacetime indices, we use Greek characters, such as $\mu, \nu = 0, 1, 2, 3$, while for spatial indices we use Latin characters, such as $i, j = 1, 2, 3$.

2.1 Basics of gravitational-wave theory

In one of the great scientific breakthroughs of the 20th century, Albert Einstein has followed up his special theory of relativity [217] by formulating the general theory of relativity [1], also simply called general relativity (GR), which forms humanity's modern understanding of gravity and the universe. Its predictions have been verified by numerous experiments [5, 218–221] in the more than 100 years since its first publication and it has allowed physicists to understand numerous physical phenomena, as well as predict new ones.

This chapter covers the basics of GR and GW theory crucial for our application, mainly following [222]. We describe the basic principle of GW detectors, their noise distribution and the basic methods used to analyze the measured data, in order to give an introduction to the research projects described in Chapters 4 and 5.

2.1.1 General theory of relativity

The bedrock of GR is the tensor formalism of pseudo-Riemannian geometry, whose key property is coordinate invariance. Tensors are defined as multilinear mappings in the (co-)tangent spaces of a spacetime manifold. We usually work with their coordinate

representations, which must follow the basic coordinate transformation laws ($x^\mu \rightarrow x'^\mu$)

$$T'^{\alpha'_1 \dots \alpha'_m}_{\beta'_1 \dots \beta'_n} = T^{\alpha_1 \dots \alpha_m}_{\beta_1 \dots \beta_n} \frac{\partial x'^{\alpha'_1}}{\partial x^{\alpha_1}} \cdots \frac{\partial x'^{\alpha'_m}}{\partial x^{\alpha_m}} \frac{\partial x^{\beta_1}}{\partial x'^{\beta'_1}} \cdots \frac{\partial x^{\beta_n}}{\partial x'^{\beta'_n}} . \quad (2.1)$$

The $\frac{\partial x'^{\alpha'}}{\partial x^\alpha}$ and $\frac{\partial x^\alpha}{\partial x'^{\alpha'}}$ are coordinate representations of the transition matrix on the tangent and cotangent spaces of the manifold at a given point.

The main object used to fully describe the geometry and evolution of a spacetime is the *metric tensor* $g_{\mu\nu}$. Its inverse is denoted $g^{\alpha\beta}$, defined by the equation

$$g^{\alpha\mu} g_{\mu\nu} = \delta_\nu^\alpha , \quad (2.2)$$

using the Kronecker tensor

$$\delta_\nu^\alpha := \begin{cases} 1 & \alpha = \nu , \\ 0 & \alpha \neq \nu . \end{cases} \quad (2.3)$$

In a physical spacetime, the metric is required to be invertible, i.e. Eq. (2.2) must have a solution for $g^{\alpha\beta}$. It then follows that the solution is unique.

We make use of Einstein's summation convention: an implied summation over each spacetime index which appears once as covariant (lower index) and once contravariant (upper index), e.g.

$$T^{\alpha\mu}{}_\beta U_{\mu\kappa} := \sum_{\mu=0}^3 T^{\alpha\mu}{}_\beta U_{\mu\kappa} , \quad (2.4)$$

where $T^{\alpha\mu}{}_\beta$ and $U_{\mu\kappa}$ are arbitrary tensors.

The Christoffel symbols are defined as

$$\Gamma^\alpha_{\mu\nu} := \frac{1}{2} g^{\alpha\beta} (g_{\beta\mu,\nu} + g_{\nu\beta,\mu} - g_{\mu\nu,\beta}) , \quad (2.5)$$

using the convention of denoting partial derivatives of tensor fields as

$$A^{\alpha_1 \dots \alpha_m}_{\beta_1 \dots \beta_n, \nu} := \frac{\partial A^{\alpha_1 \dots \alpha_m}_{\beta_1 \dots \beta_n}}{\partial x^\nu} . \quad (2.6)$$

The Christoffel symbols are not tensors, and in general, neither are partial derivatives of tensors! Their purpose is the definition of the covariant derivative of a tensor

$$\begin{aligned} T^{\alpha_1 \dots \alpha_m}_{\beta_1 \dots \beta_n; \mu} &:= T^{\alpha_1 \dots \alpha_m}_{\beta_1 \dots \beta_n, \mu} \\ &+ \Gamma^{\alpha_1}_{\mu\nu} T^{\nu \alpha_2 \dots \alpha_m}_{\beta_1 \dots \beta_n} \dots + \Gamma^{\alpha_m}_{\mu\nu} T^{\alpha_1 \dots \alpha_{m-1} \nu}_{\beta_1 \dots \beta_n} \\ &- \Gamma^{\nu}_{\mu\beta_1} T^{\alpha_1 \dots \alpha_m}_{\nu \beta_2 \dots \beta_n} \dots - \Gamma^{\nu}_{\mu\beta_n} T^{\alpha_1 \dots \alpha_m}_{\beta_1 \dots \beta_{n-1} \nu} , \end{aligned} \quad (2.7)$$

which, despite consisting of non-tensor terms, is a tensor. In a similar manner, the motion of a test particle follows the *geodesic equation*

$$\frac{d^2 x^\mu}{d\tau^2} + \Gamma^\mu_{\alpha\beta} \frac{dx^\alpha}{d\tau} \frac{dx^\beta}{d\tau} = 0 . \quad (2.8)$$

The Riemann curvature tensor $R^\rho_{\sigma\mu\nu}$ is then defined using the commutator

$$V_{\sigma;\mu\nu} - V_{\sigma;\nu\mu} = V_\rho \underbrace{\left(\Gamma^\rho_{\nu\sigma, \mu} - \Gamma^\rho_{\mu\sigma, \nu} + \Gamma^\rho_{\mu\lambda} \Gamma^\lambda_{\nu\sigma} - \Gamma^\rho_{\nu\lambda} \Gamma^\lambda_{\mu\sigma} \right)}_{R^\rho_{\sigma\mu\nu} :=} , \quad (2.9)$$

and the Ricci tensor $R_{\mu\nu}$ and scalar R as its contractions

$$R_{\mu\nu} := R^\alpha_{\mu\alpha\nu} , \quad R := g^{\mu\nu} R_{\mu\nu} . \quad (2.10)$$

Using these curvature tensors, Albert Einstein formulated the gravitational field equation, as a result of several assumptions, and matching the weak field regime to Newtonian gravity, as [1, 222]

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi T_{\mu\nu} , \quad (2.11)$$

where $T_{\mu\nu}$ is the *stress-energy tensor*. It contains information on the matter in the spacetime as well as fields which carry energy and momentum, e.g. the electromagnetic field [223]. When $T_{\mu\nu} = 0$, we are dealing with a *vacuum spacetime*, otherwise it serves as the source term in the equation and through its effect on spacetime geometry, it leads to gravitational force.

2.1.2 Linearized Einstein equation

A key part of gravitational field dynamics is the slow body motion and weak field regime. As we will see later, it is this regime which determines how GWs move through space and time from the source, eventually reach Earth and may trigger a man-made

detector. First, we assume our spacetime to be a perturbation of flat spacetime:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} , \quad (2.12)$$

where

$$\eta_{\mu\nu} = \eta^{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.13)$$

is the Minkowski metric and $h_{\mu\nu}$ is a small deviation. We can obtain the inverse to first order as

$$g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu} + o(h) , \quad h^{\mu\nu} := \eta^{\mu\alpha} \eta^{\nu\beta} h_{\alpha\beta} . \quad (2.14)$$

Only taking $\mathcal{O}(h)$ terms in Eq. (2.11), we obtain the linearized Einstein equation

$$h^{\sigma}_{\mu,\sigma\nu} + h^{\sigma}_{\nu,\sigma\mu} - \square h_{\mu\nu} - h_{,\mu\nu} - (h^{\sigma\alpha}_{,\sigma\alpha} - \square h) \eta_{\mu\nu} = 16\pi T_{\mu\nu} , \quad (2.15)$$

using the trace $h := \eta^{\mu\nu} h_{\mu\nu}$ and the d'Alembert operator

$$\square := \eta^{\kappa\lambda} \frac{\partial}{\partial x^{\kappa}} \frac{\partial}{\partial x^{\lambda}} . \quad (2.16)$$

We further define the trace reversal operation as

$$\bar{h}_{\mu\nu} := h_{\mu\nu} - \frac{1}{2} h \eta_{\mu\nu} , \quad (2.17)$$

which is a 1-to-1 map and it simplifies Eq. (2.15) to

$$\bar{h}^{\sigma}_{\mu,\sigma\nu} + \bar{h}^{\sigma}_{\nu,\sigma\mu} - \bar{h}^{\sigma\alpha}_{,\sigma\alpha} \eta_{\mu\nu} - \square \bar{h}_{\mu\nu} = 16\pi T_{\mu\nu} . \quad (2.18)$$

Let us now introduce a coordinate transformation of the form

$$x'^{\mu} := x^{\mu} + \xi^{\mu}(x) , \quad \xi = \mathcal{O}(h) , \quad (2.19a)$$

$$\Rightarrow h'_{\mu\nu} = h_{\mu\nu} - \xi_{\mu,\nu} - \xi_{\nu,\mu} . \quad (2.19b)$$

In this context, such a transformation is also called a gauge transformation, as the $h'_{\mu\nu}$ describes a spacetime identical to that of $h_{\mu\nu}$, only in a different set of coordinates. Choosing a ξ such that $\square \xi^{\mu} = \bar{h}^{\mu\alpha}_{,\alpha}$, which always has a solution for ξ , it holds

$\bar{h}'^{\sigma}_{\mu,\sigma\nu} = 0$. Therefore, we can fix the *Lorentz gauge* [222]

$$\bar{h}^{\sigma}_{\mu,\sigma\nu} = 0 , \quad (2.20)$$

in which case Eq. (2.18) simplifies to the well-known wave equation

$$\square \bar{h}_{\mu\nu} = -16\pi T_{\mu\nu} . \quad (2.21)$$

2.1.3 Plane wave solution

We start by solving the homogeneous equation, i.e. Eq. (2.21) with $T_{\mu\nu} = 0$:

$$\square \bar{h}_{\mu\nu} = 0 . \quad (2.22)$$

As with any other wave equation, we must examine the plane wave solutions

$$\bar{h}_{\mu\nu} = \text{Re} [A_{\mu\nu} \exp (ik_{\alpha}x^{\alpha})] , \quad (2.23)$$

where the components of $A_{\mu\nu}$ and k_{μ} are constants, and $A_{\mu\nu}$ is a symmetrical complex-valued tensor. In order for this to be a solution, the following must hold:

$$A_{\mu\nu}k^{\nu} = 0 , \quad k^{\alpha}k_{\alpha} = 0 . \quad (2.24)$$

This solution represents a plane GW: a perturbation of the spacetime geometry, propagating at the speed of light (as k^{μ} is a null vector) in a single direction.

To investigate its behavior closer, let us choose a coordinate system such that the wave propagates along the z direction, i.e. $k^{\mu} = (k, 0, 0, k)$. Furthermore, we use the transverse-traceless gauge

$$\bar{h}_{\mu\nu}u^{\nu} = 0 , \quad \bar{h}^{\mu}_{\mu} = 0 , \quad (2.25)$$

where the u^{μ} is a vector field whose components are constants. Let us choose this as $u^{\mu} = (1, 0, 0, 0)$. A detailed discussion of this gauge freedom goes beyond the scope of this text, see, e.g., [222].

At this point, there are only two independent components in the amplitude tensor: $A_{xx} = -A_{yy}$ and $A_{xy} = A_{yx}$, corresponding to two independent polarizations of GWs. For reasons detailed below, they are called “plus” and “cross” polarizations and denoted

the following way:

$$A_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A_+ & A_\times & 0 \\ 0 & A_\times & -A_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.26)$$

Let us imagine a ring of test particles in a Minkowski spacetime, with zero initial velocity. They remain stationary until a GW h passes through them. Obviously $\dot{x}^j = 0$, $j = 1, 2, 3$ is still a solution to the geodesic equation to first order in h , as

$$\Gamma^j_{00} = \frac{1}{2} g^{j\mu} (2g_{\mu 0,0} - g_{00,\mu}) = \frac{1}{2} \eta^{j\mu} (2h_{\mu 0,0} - h_{00,\mu}) + o(h) = o(h), \quad (2.27a)$$

$$\Rightarrow \ddot{x}^j = -\Gamma^j_{\mu\nu} \dot{x}^\mu \dot{x}^\nu = -\Gamma^j_{00} (\dot{x}^0)^2 = 0. \quad (2.27b)$$

Therefore the test particles remain stationary in the transverse-traceless frame. However, due to the changes in the metric, the distances between any given pair of points on the ring may change in time, as well as the angles in a triangle formed by any three points on the ring.

To visualize these geometrical effects, we make use of an equivalent description of these spatial structure changes, which consists of the same ring of test particles in the stationary Minkowski spacetime. The ring, however, is being deformed by the wave through the expression

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \sqrt{\begin{pmatrix} 1+h_+ & h_\times \\ h_\times & 1-h_+ \end{pmatrix}} \begin{pmatrix} x \\ y \end{pmatrix} \approx \begin{pmatrix} 1 + \frac{h_+}{2} & \frac{h_\times}{2} \\ \frac{h_\times}{2} & 1 - \frac{h_+}{2} \end{pmatrix}. \quad (2.28)$$

The approximation in Eq. (2.28) holds when $h = \sqrt{h_+^2 + h_\times^2} \ll 1$. This map deforms the ring with radius r into an ellipse with semimajor and semiminor axis lengths

$$a = r\sqrt{1+h} \approx r \left(1 + \frac{h}{2}\right), \quad b = r\sqrt{1-h} \approx r \left(1 - \frac{h}{2}\right), \quad (2.29)$$

respectively. In the simplest case $h_\times = 0$, the ring is extended in the x direction and contracted in the y direction when $h_+ > 0$, and vice versa when $h_+ < 0$. When $h_+ = 0$, it can be shown in a similar manner that the h_\times polarization does the same along axes at 45° angles with respect to the x and y axes. This effect is demonstrated in Fig. 2.1 on the example of a sine wave and is the reason for the polarizations being denoted $+$ and \times .

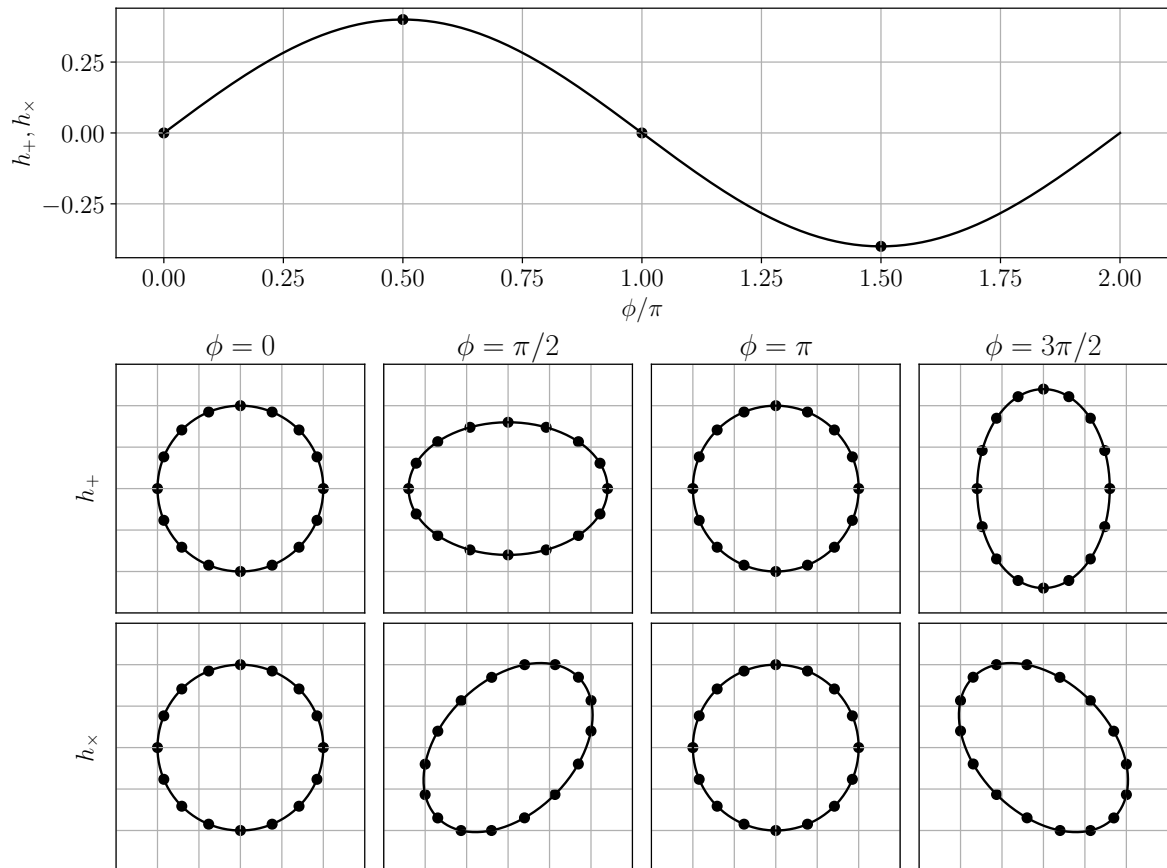


FIGURE 2.1: Illustration of the effect of the GW polarizations h_+ and h_\times passing through matter. The top panel shows the phase of a hypothetical sine waveform with amplitude 0.4. The bottom two rows show the effect of the wave passing through a circle of 16 stationary test masses in the plane perpendicular to the wave vector \vec{k} of the GW in cases where the polarization is h_+ and h_\times , respectively. The four columns show the effect of the wave at the four phases highlighted in the top panel. The linear approximation of Eq. (2.28) is used.

2.1.4 Gravitational wave radiation

Let us now assume an isolated system of masses m_A which move with respect to each other. In analogy with electromagnetic theory, we define its *dipole moment* and its *traceless quadrupole moment* as

$$\mathbf{d} := \sum_A m_A \mathbf{x}_A, \quad \mathcal{I}_{jk} := \sum_A m_A \left(x_{Aj} x_{Ak} - \frac{1}{3} \delta_{jk} r_A^2 \right), \quad (2.30)$$

respectively, or in the continuous case, $\sum_A m_A f(\mathbf{x}_A) \mapsto \int dm f(\mathbf{x})$.

The solution to Eq. (2.21) can be expressed using the Green's function approach. Far from the source, it simplifies to the *quadrupole formula* [222]

$$\bar{h}_{ij}(t, r) = \frac{2}{r} \ddot{I}_{ij}(t - r) , \quad (2.31)$$

and the total energy radiated through GWs in all directions per unit time is

$$L_{\text{gw}} = \frac{1}{5} \sum_{j=1}^3 \sum_{k=1}^3 \langle (\ddot{I}_{jk})^2 \rangle . \quad (2.32)$$

As a consequence, a pair of point masses m_1, m_2 in a quasi-circular orbit can be modeled simply by Newtonian mechanics with an energy loss rate following Eq. (2.32). Their separation evolves as

$$a(t) = a_0 \left(1 - \frac{t}{\tau_0} \right)^{1/4} , \quad (2.33)$$

where a_0 is the initial separation, and they merge in a finite time [222]

$$\tau_0 = \frac{5a_0^4}{256m_1m_2(m_1 + m_2)} . \quad (2.34)$$

The quadrupole formula (2.31) above describes gravitational radiation accurately under the assumption of low density and velocities of the source. However, these assumptions do not hold for currently observable GW sources close to merger. To model them, more intricate techniques are required. Note that the dipole moment \mathbf{d} does not contribute to GW radiation; this is related to momentum conservation and the spin-2 nature of the gravitational field [1, 222, 224, 225].

2.2 Compact binary coalescence

While the Einstein equation in Eq. (2.11) is complicated and non-linear and in many applications requires numerical solvers, its solution representing the stationary gravitational field of a simple point particle of mass M has been derived and published by Karl Schwarzschild barely a month after the publication of GR [226]. Its key feature is the sphere at the radial coordinate value $r = 2M$ called the *event horizon*: a barrier through which no null or timelike curve can exit the central object. An object described by this metric is called a black hole (BH).

Through the same formation mechanism as stellar BHs (gravitational collapse of a massive star [227, 228]), an object may arise whose gravity is sufficient to overcome the electron degeneracy pressure, but not the neutron degeneracy pressure, which is the effect of the Fermi exclusion principle of the electron and neutron gas, respectively. This object is called a *neutron star (NS)*. The event horizon does not form and matter from any part of the object can, in principle, leave its boundaries [229].

A significant topic of discussion are the masses of compact objects. The Tolman-Oppenheimer-Volkoff limit $M_{\text{TOV}} \approx 2.1M_{\odot}$ [230] is the upper bound on the mass of a NS. The general consensus is that any NS above this mass would collapse into a BH. For a BH, there is no specific limit based on its stability. However, stellar physics has set an approximate range of masses for stellar BH formation at $[5M_{\odot}, 50M_{\odot}]$ [22].

In principle, any system of accelerated masses radiates GWs following Eq. (2.31) in the Newtonian limit. However, most are too weak to be detected by current detectors on Earth. We focus on a particular type of event: a compact binary coalescence (CBC). It is the final evolution stage of a binary composed of two compact objects, which gradually loses energy through GW radiation and eventually merges into a single compact object, typically a BH, and the only confidently detected GW source type to date [72]. Specifically, the components of these binaries are BHs and/or NSs, as binaries with a white dwarf (which is another compact object type) as one or both of their components are not sufficiently loud to be relevant to contemporary GW research [231]. This leaves us with 3 types of CBC events: binary black hole (BBH) mergers, binary neutron star (BNS) mergers, and black hole-neutron star (BHNS) mergers.

BNS and BHNS events come with a chance of a detectable electromagnetic counterpart, allowing for *multi-messenger observations*, providing additional information about the merger and allowing to better understand the microphysics of NS matter [15, 201]. In addition, some elements which naturally exist on Earth and without which life could not exist, are produced by BNS mergers, making GW observations crucial to better understand the universe and its history [232, 233].

In this thesis, we focus on BBH events as a suitable entry point into machine learning-based GW data analysis for their relative simplicity, as no knowledge of nuclear matter properties is required. Furthermore, due to the steeply rising noise characteristic of contemporary GW detectors towards lower frequencies (see Sec. 2.3), each compact binary spends a limited time in a detector's sensitive range. For the purpose of a search it is sufficient to model this section of the inspiral, which is shorter in the

category	symbol	name	distribution
intrinsic	$m_1 \geq m_2$	component masses	
	χ_1, χ_2	dimensionless spins	
extrinsic	Φ_0	coalescence phase	$\Phi_0 \in [0, 2\pi)$
	Ψ	polarization	$\Psi \in [0, 2\pi)$
	ι	inclination	$\cos \iota \in [-1, 1]$
	θ	declination	$\sin \theta \in [-1, 1]$
	φ	right ascension	$\varphi \in [-\pi, \pi)$
	d_L	luminosity distance	$d_L^2 \in \mathbb{R}^+$
	t_c	time of merger	

TABLE 2.1: Summary of parameters required to describe a BBH merger as seen in a terrestrial detector. The last column shows the corresponding astrophysical distribution where one exists. The intervals refer to a uniform distribution.

case of a BBH merger than CBCs with a NS component, making them more suitable for ML-driven analysis.

To fully describe a BBH merger with respect to the Earth, we require a total of 15 parameters, summarized in Tab. 2.1. They are divided into two categories: *intrinsic* and *extrinsic*. The former describe the physical properties of the system, while the latter specify its relative position with respect to the Earth as well as the time when the signal reaches the Earth. The intrinsic parameters are the masses and angular momenta of the two components. The dimensionless spin is defined as $\chi_i := \mathbf{S}_i/m_i^2$, $i = 1, 2$, where \mathbf{S}_i is the angular momentum of the respective binary component and m_i its mass. Analysis of the Kerr spacetime [222, 234] reveals that it must hold $|\chi_i| \leq 1$ for the object to be a BH and the opposite yields a spacetime considered unphysical.

The extrinsic parameters are the luminosity distance d_L , the time of merger as seen by an observer on Earth t_c , and a set of 5 angles which fully characterize the coordinate transformation between the source and the detector. In fact, these are two sets of Euler angles to match the two coordinate frames:

- source frame
 - *coalescence phase* Φ_0 : rotation around the total angular momentum vector,
 - *inclination* ι : angle between the orbital angular momentum at the reference frequency and the line-of-sight to Earth,
 - *polarization* Ψ : rotation around the line-of-sight, degenerate with the corresponding angle in the detector frame rotation,

- detector frame
 - *right ascension* φ : celestial equivalent of the longitude (rotation around the Earth’s axis),
 - *declination* θ : celestial equivalent of the latitude (angle between the equatorial plane and the line-of-sight),
 - *polarization* Ψ : rotation around the line-of-sight, degenerate with the corresponding angle in the source frame rotation.

To fully model a CBC and compute the corresponding waveform, the following methods are used:

- Numerical relativity (NR): a full discretization of the spacetime and numerical evolution of $g_{\mu\nu}$ through the inspiral, merger and ringdown. Most accurate in the late stages but highly computationally demanding. [235–242]
- Post-Newtonian expansion: Taylor expansion of the equations of motion in powers of v^2/c^2 , v being the characteristic velocity of the system. Efficient but they are asymptotic series which become inaccurate close to merger. [243–247]
- Effective-one-body (EOB) models: based on systematic resummations of post-Newtonian expansion methods, a Hamiltonian is constructed which reduces the system to the motion of a test particle in an external metric. Complete with a phenomenological description of the merger and ringdown, very accurate and efficient. [248–254]
- Phenomenological models: a simple model fit to hybrid EOB-NR waveforms. Very accurate and efficient, they are the core of most contemporary conventional analyses, along with EOB models. [255–260]
- NR surrogate models: ML-based models trained on NR waveforms. Accurate and efficient, especially in highly precessing waveform. [261–270].

An example waveform generated by the phenomenological model IMRPhenomD [259] is shown in Fig. 2.2. The characteristic shape of a gradual rise in both frequency and amplitude, culminating and then quickly fading is called a *chirp*, and is a typical shape of all BBH waveforms [271]. In case of BNS and BHNS mergers, the post-merger section is much more pronounced due to the matter remaining outside the remnant [272, 273].

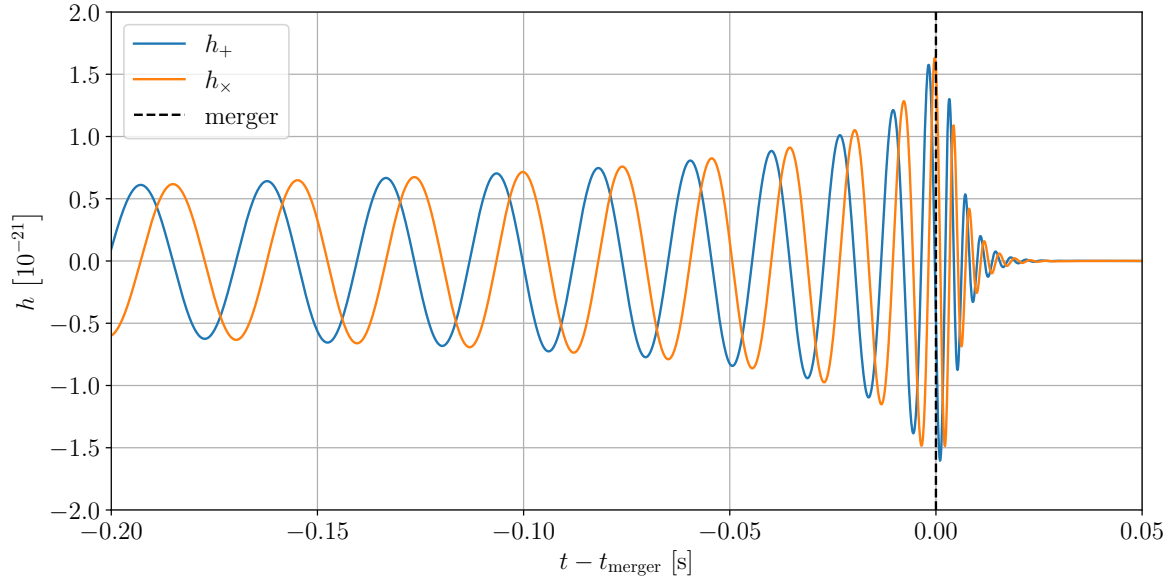


FIGURE 2.2: An example waveform from a BBH merger with component masses $36M_{\odot}$ and $29M_{\odot}$ at a luminosity distance of 470 Mpc, with zero spins, roughly corresponding to the GW150914 event [6]. The waveform has been generated by the phenomenological model IMRPhenomD [259]. The dashed black line marks the time of the merger, on its left side is the inspiral phase, and on the right side the ringdown.

2.3 Interferometric gravitational wave detectors

As shown in Sec. 2.1, a GW passing through several test masses contracts and dilates the spacetime between these masses. This is used in interferometric GW detectors, which in their essence replicate the design of a classical Michelson interferometer, as the variation in distance between the beam splitter and the mirrors introduces a time-dependent phase shift. Therefore, the light intensity seen by the photodetector carries the GW signal [274].

Fig. 2.3 shows a simplified schematic of Advanced LIGO detectors. Assuming a coordinate system with x and y axes along the detector arms and a GW passing along the z axis orthogonal to the plane of the detector, a GW will extend and contract its arms as demonstrated in Fig. 2.1. In this assumed geometrical arrangement, it is insensitive to the \times polarization, as the lengths of both arms change by the same amount at the same time, introducing zero phase shift. On the other hand, the $+$ polarization extends one arm while contracting the other by the same amount, and introduces the highest possible phase shift.

This coordinate system is generally not aligned with the coordinate system of the source, and therefore it is necessary to perform a coordinate transformation described

by the three Euler angles θ , φ and Ψ defined above. The final signal observed in the detector is

$$h(t) = F_+(\theta, \varphi, \Psi) h_+(t) + F_\times(\theta, \varphi, \Psi) h_\times(t) , \quad (2.35)$$

$F_+(\theta, \varphi, \Psi)$ and $F_\times(\theta, \varphi, \Psi)$ being the *antenna patterns* of the detector, and $h_+(t)$ and $h_\times(t)$ the two independent polarizations of the given waveform. For further details, we refer the reader to [275].

The measured strain \mathbf{s} contains noise, which severely limits our detection potential. Elaborate data analysis techniques are required to extract signals from noisy data. To cover the basics, we follow the guideline of [50].

On timescales on the order of hours, the noise in GW detectors can be described as colored, stationary and Gaussian. In the frequency domain, it is uncorrelated, Gaussian with zero mean, and can be described by the probability density function

$$p(\mathbf{n}) = A \cdot \exp\left(-\frac{1}{2}(\mathbf{n}|\mathbf{n})\right) , \quad (2.36a)$$

$$(\mathbf{a}|\mathbf{b}) := 2 \int_0^\infty \frac{\tilde{a}(f) \tilde{b}^*(f) + \text{c.c.}}{S_n(f)} df , \quad (2.36b)$$

where $(\cdot|\cdot)$ is the *noise-weighted inner product* and $S_n(f)$, which is called the power spectral density (PSD), describes the noise variance at different frequencies. The noise is often described in an equivalent manner using the amplitude spectral density (ASD) $\sqrt{S_n(f)}$. Over longer timescales, however, one can observe the PSD gradually change, which we call *PSD drift*. Finally, A is a normalization constant such that

$$\int p(\mathbf{n}) d\mathbf{n} = 1 . \quad (2.37)$$

The shape of a PSD/ASD is given by multiple effects, the most significant being [277, 278]:

- seismic noise: mirror motion due to ground vibrations, earthquakes, etc.,
- thermal noise: due to fluctuations of atoms in the mirrors and their suspensions,
- quantum noise: due to the discrete nature of light,
- gas noise: due to residual gas molecules in the vacuum tubes,
- charging noise: due to interactions of static electric charges on the mirrors with the metals of the tubes and supports.

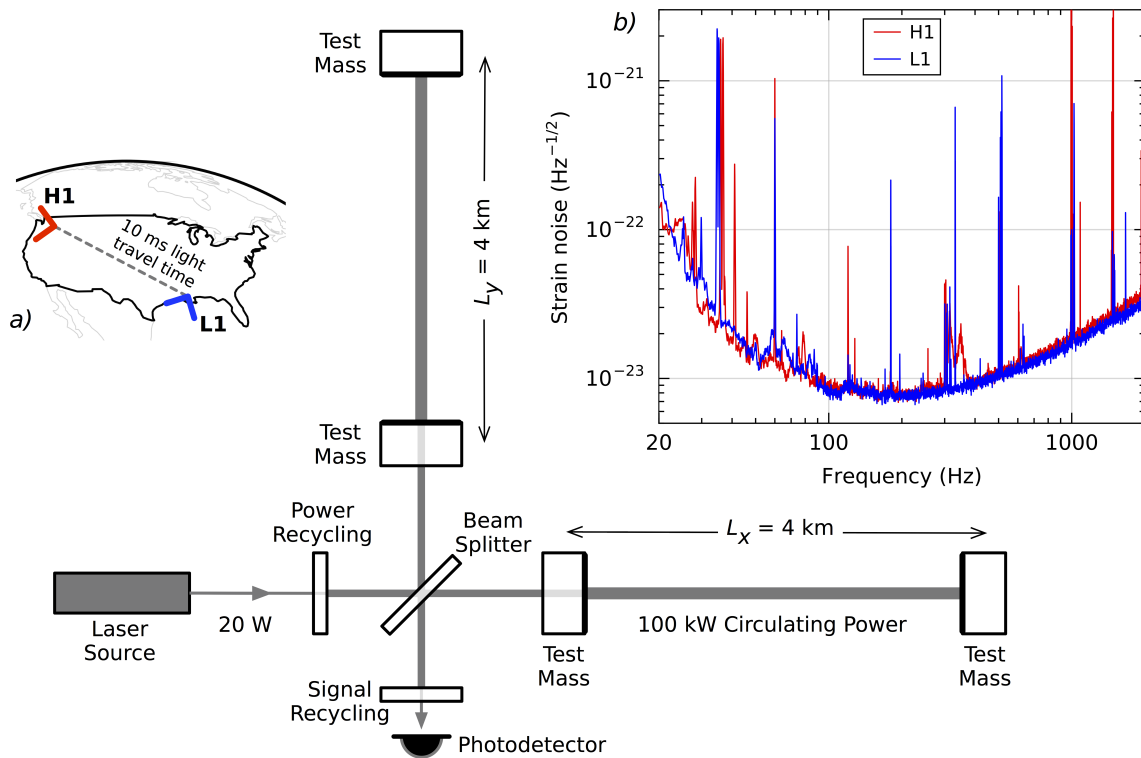


FIGURE 2.3: Simplified diagram of an Advanced LIGO detector (not to scale). A GW propagating orthogonally to the detector plane and linearly polarized parallel to the 4-km optical cavities will have the effect of lengthening one 4-km arm and shortening the other during one half-cycle of the wave; these length changes are reversed during the other half-cycle. The output photodetector records these differential cavity length variations. While a detector's directional response is maximal for this case, it is still significant for most other angles of incidence or polarizations (GWs propagate freely through the Earth). *Inset (a)*: Location and orientation of the LIGO detectors at Hanford, Washington (H1) and Livingston, Louisiana (L1). *Inset (b)*: The instrument noise for each detector near the time of the signal detection; this is an amplitude spectral density, expressed in terms of equivalent GW strain amplitude. The sensitivity is limited by photon shot noise at frequencies above 150 Hz, and by a superposition of other noise sources at lower frequencies [276]. Narrow-band features include calibration lines (33-38, 330, and 1080 Hz), vibrational modes of suspension fibers (500 Hz and harmonics), and 60 Hz electric power grid harmonics. Figure and caption taken from [5].

Example ASDs are shown in inset b) of Fig. 2.3; more specifically, they correspond to the noise observed in LIGO detectors around the GW150914 event. They contain all the characteristic properties of contemporary detector ASDs: steep rise at low and high frequencies, minimum around 200 Hz, narrow peaks at various frequencies.

One can visualize strain data from GW detectors using a spectrogram. The signal is split into chunks (usually overlapping) and the frequency spectrum of the individual chunks is computed and plotted using a colormap in the time-frequency space, taking the time of the corresponding chunk on the x axis and the frequency of the Fourier transform on the y axis.

The basic method relies on the Fourier transform. In GW science, it is more common to use the Q-transform instead [279], which performs a very similar task while covering the frequency space on a logarithmic scale more uniformly than the Fourier transform. Fig. 2.4 shows, as an example, the spectrogram of the GW150914 event as recorded by the Hanford detector [5]. It clearly shows the typical chirp of a CBC waveform as a patch of high power at a frequency growing with time, the random weaker spots in the background representing the detector noise.

The noise distribution possesses another feature, namely noise transients, also called *glitches*. They appear as a large amount of excess power and some can easily be confused for a GW chirp signal. This strongly reduces detector sensitivity in segments where glitches are present. In LIGO-Virgo experiments, a large set of auxiliary channels is employed to detect anomalous behavior and flag segments which likely contain a glitch, these are then excluded from most searches [280–284] using tools such as Omicron [285]. Remaining ones are excluded through *veto*es [52, 53, 286, 287]. The daily median glitch rates in the H1, L1, and V1 detectors during the O3a and O3b observing runs lie between 0.29 min^{-1} and 1.17 min^{-1} , which corresponds to one glitch in just over three minutes in the former case, and less than a minute in the latter [72, 288]. As conventional searches operate at false alarm rates $< 0.5 \text{ month}^{-1}$, this makes handling glitches a crucial part of GW search algorithms.

Due to the various mechanisms which cause glitches to occur, they come in many shapes, the understanding of which can be helpful in cleaning detector data, and ML-based methods have been used to classify them [289–293]. For illustration, we attach a set of spectrograms of different glitch types in Fig. 2.5. In this thesis, knowledge of glitch morphology is not required, as the project of Ch. 4 and a part of Ch. 5 deal with a Gaussian noise model, and in the remainder the focus is on ML models learning the glitch morphology from data without information on their types.

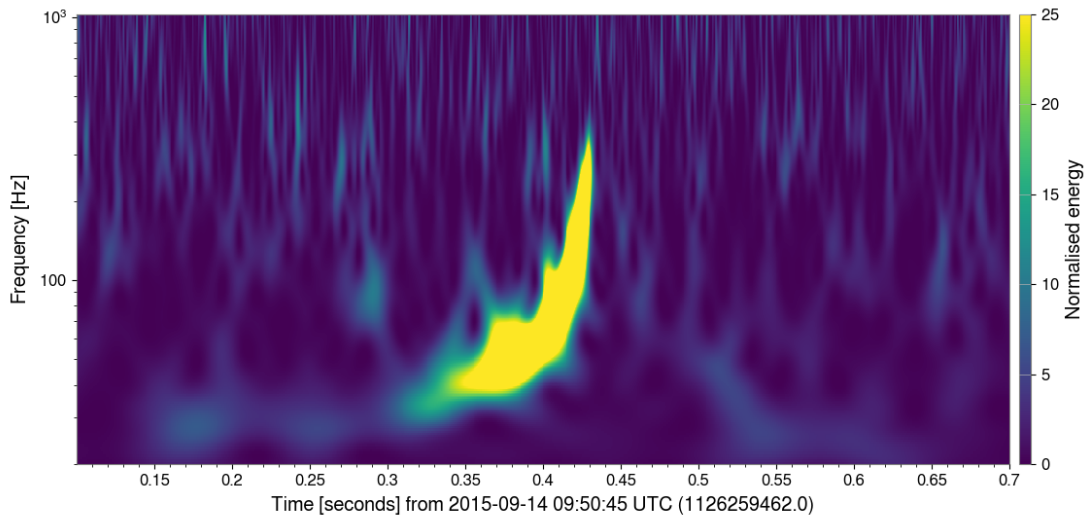


FIGURE 2.4: Q-transform plot of the GW150914 event [5] as seen by the LIGO detector in Hanford, Washington. Figure taken from GWOSC [294].

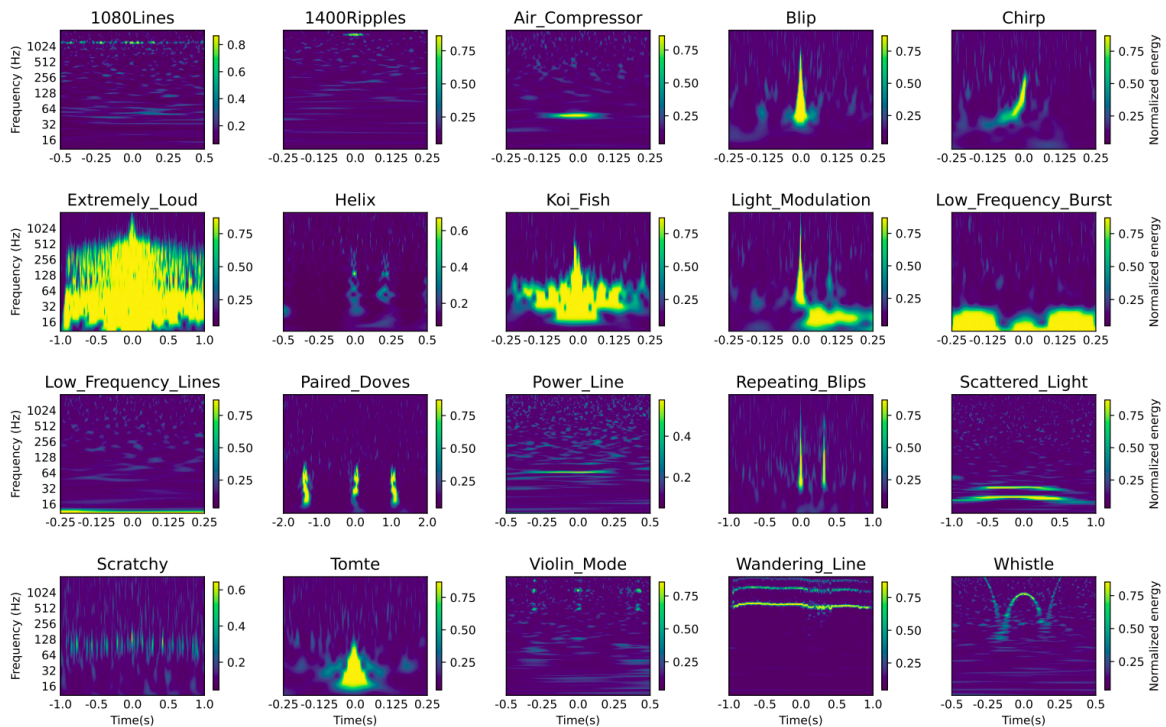


FIGURE 2.5: Representative spectrograms for all glitch classes contained in the GravitySpy dataset [289]. Figure taken from [295].

2.4 Detector data analysis

2.4.1 Matched filtering

A GW search is expected to produce a set of candidate events, specified by a time and a ranking statistic. The most standard searches are based on *matched filtering*: using the noise-weighted inner product defined in Eq. (2.36b) to find matches between the detector data stream and a template bank consisting of generated waveforms. This section comprises a superficial overview of the methods relevant to our work, following mainly [50].

Let \mathcal{H}_1 be the hypothesis that a segment of data \mathbf{s} from a single detector includes a specific GW waveform \mathbf{h}

$$\mathbf{s} = \mathbf{n} + \mathbf{h} , \quad (2.38)$$

\mathbf{n} being the background noise following Eq. (2.36a). In addition, let \mathcal{H}_0 be the hypothesis that \mathbf{s} is composed purely of noise \mathbf{n} . One can then use the Bayes theorem [50, 296] and the expected noise distribution

$$p(\mathcal{H}_1|\mathbf{s}) = \frac{p(\mathbf{s}|\mathcal{H}_1)p(\mathcal{H}_1)}{p(\mathbf{s}|\mathcal{H}_1)p(\mathcal{H}_1) + p(\mathbf{s}|\mathcal{H}_0)p(\mathcal{H}_0)} = \left[1 + \frac{p(\mathcal{H}_0)}{p(\mathcal{H}_1)\Lambda(\mathbf{s})} \right]^{-1} , \quad (2.39)$$

using the *likelihood ratio*

$$\Lambda(\mathbf{s}) := \frac{p(\mathbf{s}|\mathcal{H}_1)}{p(\mathbf{s}|\mathcal{H}_0)} . \quad (2.40)$$

As $p(\mathcal{H}_1|\mathbf{s})$ is a purely growing function of $\Lambda(\mathbf{s})$, we can use the likelihood ratio as an equivalent detection ranking statistic. Furthermore, let us assume that $\mathbf{h} = \mathbf{h}(\boldsymbol{\theta})$, $\boldsymbol{\theta}$ being the source parameters, in case of a BBH merger the ones described in Sec. 2.2. In the context of a search, they are also randomly distributed, and thus it holds

$$p(\mathbf{s}|\mathcal{H}_1) = A \int \exp \left[-\frac{1}{2} (\mathbf{s} - \mathbf{h}(\boldsymbol{\theta}) | \mathbf{s} - \mathbf{h}(\boldsymbol{\theta})) \right] p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (2.41a)$$

$$\Rightarrow \Lambda(\mathbf{s}) = \int \exp \left[\frac{1}{2} \rho^2(\mathbf{s}, \boldsymbol{\theta}) \right] p(\boldsymbol{\theta}) d\boldsymbol{\theta} , \quad (2.41b)$$

$$\rho^2(\mathbf{s}, \boldsymbol{\theta}) := (2\mathbf{s} - \mathbf{h}(\boldsymbol{\theta}) | \mathbf{h}(\boldsymbol{\theta})) . \quad (2.41c)$$

The integrand of Eq. (2.41b) is generally sharply peaked around its maximum [50]. Therefore, this maximum is expected to be a good approximation, up to a constant

rescaling [50]. This is equivalent to taking a maximum of the argument of the exponential

$$\rho_{\text{MF}}^2 := \max_{\boldsymbol{\theta}} \rho^2(\mathbf{s}, \boldsymbol{\theta}) . \quad (2.42)$$

Typically, ρ_{MF} is called the *matched-filter signal-to-noise ratio (SNR)*, and is used as a ranking statistic in detection.

As the extrinsic parameters of a signal are unknown prior to detection, and therefore the antenna patterns as well as the amplitude are unknown too, a search looks for

$$\mathbf{h}(\boldsymbol{\theta}) = p\mathbf{h}_+(\boldsymbol{\mu}) + q\mathbf{h}_\times(\boldsymbol{\mu}) , \quad p, q \in \mathbb{R} , \quad (2.43a)$$

$$(\mathbf{h}_+|\mathbf{h}_+) = (\mathbf{h}_\times|\mathbf{h}_\times) = 1 , \quad (\mathbf{h}_+|\mathbf{h}_\times) = 0 , \quad (2.43b)$$

where the p and q parameters encompass the dependence on all extrinsic parameters besides the merger time, and $\boldsymbol{\mu}$ are the intrinsic parameters. When dealing with waveforms without higher modes in a single detector, these are degenerate and the full morphology can be obtained by varying $\boldsymbol{\mu}$ and the coefficients p and q :

$$\rho_{\text{MF}}^2(\mathbf{s}) = \max_{\boldsymbol{\mu}} \max_{p,q} (2\mathbf{s} - p\mathbf{h}_+(\boldsymbol{\mu}) - q\mathbf{h}_\times(\boldsymbol{\mu}) | p\mathbf{h}_+(\boldsymbol{\mu}) + q\mathbf{h}_\times(\boldsymbol{\mu})) . \quad (2.44)$$

The optimization over p and q can be done analytically as simply as

$$p_{\text{max}} = (\mathbf{s}|\mathbf{h}_+(\boldsymbol{\mu})) , \quad q_{\text{max}} = (\mathbf{s}|\mathbf{h}_\times(\boldsymbol{\mu})) , \quad (2.45a)$$

$$\rho_{\text{MF}}^2(\mathbf{s}) = \max_{\boldsymbol{\mu}} [(\mathbf{s}|\mathbf{h}_+(\boldsymbol{\mu}))^2 + (\mathbf{s}|\mathbf{h}_\times(\boldsymbol{\mu}))^2] . \quad (2.45b)$$

Furthermore, $(\mathbf{s}|\mathbf{h}_n)$ with \mathbf{h}_n a normalized waveform is called a *matched filter* and is a crucial element of parameter estimation algorithms.

A template bank is a collection of normalized waveforms, covering densely the intrinsic parameter space $\boldsymbol{\mu}$. An algorithm searching for events then computes the ρ_{MF} for each waveform from the template bank, shifting it through the entire dataset under examination to account for an unknown event time. This can be done using a convolution, computed efficiently using fast Fourier transform algorithms, which also motivates the use of CNN models (see Sec. 3.3) later in this thesis.

We also define the *optimal SNR* of a waveform \mathbf{h} as

$$\rho_{\text{opt}}^2 := (\mathbf{h}|\mathbf{h}) , \quad (2.46)$$

which deviates from the ρ_{MF} through the $(\mathbf{n}|\mathbf{h})$ terms when the template \mathbf{h} is the true injection. When dealing with a waveform outside a detector without knowledge of a background noise sample, ρ_{opt} provides an approximation of ρ_{MF} .

Furthermore, when using a network of multiple detectors with SNRs ρ_I indexed by I , let us assume the noise distributions of the individual networks to be independent

$$\rho_{\text{net}}^2 \propto \log \Lambda = \log \left(\prod_I \Lambda_I \right) = \sum_I \log \Lambda_I \propto \sum_I \rho_I^2 \quad (2.47)$$

Based on this, we define the *network SNR* [52]

$$\rho_{\text{net}}^2 := \sum_I \rho_I^2 . \quad (2.48)$$

In matched-filter based detection pipelines, in case of coincident triggers from multiple detectors, the formula above is commonly used as a ranking statistic.

One may notice that Eq. (2.36b) can be rewritten as

$$(\mathbf{a}|\mathbf{b}) = 2 \int_0^\infty \left[\tilde{a}_w(f) \tilde{b}_w^*(f) + \text{c.c.} \right] df , \quad (2.49a)$$

$$\tilde{a}_w(f) := \frac{\tilde{a}(f)}{\sqrt{S_n(f)}} . \quad (2.49b)$$

Plugging this into the noise distribution in Eq. (2.36a) we see that this transformation yields uncorrelated noise, which corresponds to a flat PSD. This way, one can transform a stream of colored data \mathbf{a} to a white stream \mathbf{a}_w , and we call this process *whitening*. At the same time, this changes the waveform morphology when injections are present in the data.

2.4.2 Search evaluation

Finally, we focus on evaluating the performance of a GW search algorithm. Let us assume a stream of test data of length T , consisting entirely of noise, and a search algorithm which produces N_f triggers, each consisting of a merger time and a ranking statistic. Denoting by $N_{f>t}$ the number of triggers whose ranking statistics exceed a detection threshold $t \in \mathbb{R}$, we define the false alarm rate (FAR)

$$\mathcal{F}(t) := \frac{N_{f>t}}{T} , \quad (2.50)$$

and we require its inverse $t(\mathcal{F})$, which returns the threshold corresponding to the desired FAR. We say a point with a ranking statistic ρ is a trigger at FAR \mathcal{F} , if

$$\rho > t(\mathcal{F}) . \quad (2.51)$$

Let $\zeta(\mathcal{F}; \mathbf{x}, \boldsymbol{\theta})$ be the probability of a search triggering at a FAR of \mathcal{F} on a noise sample injected with a waveform produced by a source, whose spatial position with respect to the Earth is \mathbf{x} and $\boldsymbol{\theta}$ are the other parameters required to describe the source. The sensitive volume of a search at FAR \mathcal{F} is then calculated as [52]

$$V(\mathcal{F}) := \int \zeta(\mathcal{F}; \mathbf{x}, \boldsymbol{\theta}) \phi(\mathbf{x}, \boldsymbol{\theta}) \, d\mathbf{x} \, d\boldsymbol{\theta} , \quad (2.52)$$

where $\phi(\mathbf{x}, \boldsymbol{\theta})$ is the distribution of injection parameters. When the injections are performed uniformly in volume up to a maximum distance d_{\max} , it can be approximated by [52]

$$V(\mathcal{F}) \approx V(d_{\max}) \frac{N_{I,\mathcal{F}}}{N_I} , \quad (2.53)$$

where $V(d_{\max})$ is the volume of a sphere with radius d_{\max} , $N_{I,\mathcal{F}}$ is the number of detected injections at a FAR of \mathcal{F} , and N_I is the total number of injections.

Finally, the *sensitive distance* is simply the radius of a sphere with volume $V(\mathcal{F})$

$$d(\mathcal{F}) := \sqrt[3]{\frac{3V(\mathcal{F})}{4\pi}} \approx d_{\max} \sqrt[3]{\frac{N_{I,\mathcal{F}}}{N_I}} . \quad (2.54)$$

We call the plot of $d(\mathcal{F})$ the *sensitivity curve*. It carries information on the performance of the search at various values of \mathcal{F} and, by extension, the threshold t . It is a common measure of production search sensitivity [52] and, in its essence, is closely related to the receiver operating characteristic [297] (see Sec. 3.1).

Chapter 3

Machine learning

In a great deal of problems in natural sciences, one must turn to approximation methods. This may amount to, e.g., expanding a function in its Taylor series and only taking its first several terms, replacing interaction potentials between different particles in a many-particle system by an effective potential, or a great multitude of other methods [298]. Function approximation is one of the most usual applications of machine learning [130, 145, 299–304].

Tom M. Mitchell broadly defined machine learning (ML) as [305]:

“A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .”

Supervised learning relies on a sufficient amount of available data in the form of input-output pairs to tune the free parameters of a model through iterative applications to the data. Under favorable conditions, this will result in the model approximating the target function well; in fact, these methods are capable of approximating an arbitrary function to an arbitrary degree of precision with a sufficiently flexible model [121, 306, 307]. Naturally, such a method also has a lot of pitfalls, which can very easily break the faithfulness of the approximation, summarized by *no free lunch theorems* [308–310].

A common example in introductory courses to ML is the classification of handwritten digits from the MNIST dataset [311]. It comprises 70 000 grayscale 28×28 -pixel images, with integer labels from 0 to 9. Simple networks trained over times on the order of a minute using a consumer GPU are capable of exceeding an accuracy of 99% in digit classification on the MNIST dataset.

ML methods are very versatile, and function approximation by supervised learning is but a small example. This thesis only uses ML for approximation. In this chapter, we introduce the basics of this method. We will mostly follow [114], which is a thorough introduction into ML methods for applications in physics, going further than function

approximation and also covering autoencoders or generative methods, for example. For more comprehensive material on ML, see [115].

3.1 Basics of machine learning

Let us start with a generic supervised learning problem. We denote the function, which we would like to approximate, f^* . It is sometimes called the *true model*. Typically, it will hold

$$f^* : \mathcal{X} \rightarrow \mathcal{Y} , \quad (3.1a)$$

$$\mathcal{X} \subset \mathbb{R}^m, \mathcal{Y} \subset \mathbb{R}^n, m, n \in \mathbb{N} . \quad (3.1b)$$

We would like to create a function which approximates f^* . To do this, we require a further four ingredients:

- **Model** To approximate the f^* declared in Eq. (3.1a), we will use a function which accepts p extra parameters

$$f : \mathcal{X} \times \mathbb{R}^p \rightarrow \mathcal{Y} . \quad (3.2)$$

The $\boldsymbol{\theta} \in \mathbb{R}^p$ is a set of parameters usually called *weights* which are tuned during the training process in a manner which improves the faithfulness of the approximation

$$f(\mathbf{x}, \boldsymbol{\theta}) \approx f^*(\mathbf{x}) . \quad (3.3)$$

The choice of a suitable model for a given problem is a complex issue. In some cases, a simple lower-order polynomial will work out well, with its coefficients as the trainable weights, while in others, it is necessary to apply a large neural networks with many weights.

- **Data** Another necessary ingredient is a set of inputs for the f^* function and its corresponding outputs, typically including noise $\boldsymbol{\sigma}$:

$$\mathbf{X} \in \mathcal{X}^N, \mathbf{Y}, \boldsymbol{\sigma} \in \mathcal{Y}^N , \quad (3.4a)$$

$$\forall i = 1, 2, \dots, N : \mathbf{Y}_i = f^*(\mathbf{X}_i) + \boldsymbol{\sigma}_i . \quad (3.4b)$$

- **Loss function** In addition, we must define a *loss function* \mathcal{C} (often also called cost, error or energy), which measures a scalar deviation between two sets of elements of the target space \mathcal{Y} . It will not only measure how well our model is performing, it is also a crucial ingredient for the training process itself.

The most common loss used for a prediction problem is the *mean squared error* loss

$$\text{MSE} : \mathbb{R}^{mn} \times \mathbb{R}^{mn} \rightarrow \mathbb{R} , \quad (3.5a)$$

$$\bar{\mathbf{Y}}, \mathbf{Y} \mapsto \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (\bar{Y}_{ij} - Y_{ij})^2 . \quad (3.5b)$$

In case of a classification problem, one would typically use, as their target space, all possible sets of probabilities for the n classes $\mathcal{Y} = \{p \in \mathbb{R}^n \mid \sum_{i=1}^n p_i = 1\}$. Then the loss function used would typically be the *Categorical Cross Entropy*

$$\text{CCE} : \mathbb{R}^{mn} \times \mathbb{R}^{mn} \rightarrow \mathbb{R} , \quad (3.6a)$$

$$\bar{\mathbf{Y}}, \mathbf{Y} \mapsto -\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \log(\bar{Y}_{ij} Y_{ij}) . \quad (3.6b)$$

In binary classification, it is often implemented to work with only one probability as input and obtain the other as a complement to one, and called the binary cross entropy (BCE).

- **Optimizer** The final basic ingredient is an algorithm which searches for extrema of functions with high-dimensional input spaces. The idea of supervised learning is tuning the weights $\boldsymbol{\theta}$ by searching for the minimum of the loss function evaluated over a large amount of input-output pairs \mathbf{X}, \mathbf{Y} :

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}), \mathbf{Y}) . \quad (3.7)$$

These are the trained network weights, and the intended approximation is

$$f(\mathbf{x}, \boldsymbol{\theta}^*) \approx f^*(\mathbf{x}) . \quad (3.8)$$

A very simple and yet extremely useful example is *linear regression*, where one searches for the best fit of a linear function to supplied data, using the L^2 loss, also

referred to as the *least squares method*. In a 1-dimensional setting, this is as simple as $f(x, \boldsymbol{\theta}) = ax + b$, $\boldsymbol{\theta} = (a, b)$, which can be optimized analytically.

In some special cases, such as the linear regression mentioned above, the expressions for the a, b values which minimize the L^2 value over multiple data points can be obtained analytically. However, when training more complicated models, this is no longer an option; due to complicated and rugged landscapes of the loss in the space of possible weights $\boldsymbol{\theta}$, one must instead turn to *gradient-descent* based methods. This requires efficient algorithms for gradient computation, which we cover in Sec. 3.3.4.

Besides the loss function used for training, often a different metric is crucial but cannot be directly optimized by the training procedure, e.g. due to discontinuities which disallow usage of gradient-descent based methods. When dealing with a binary classification problem, the model returns its degree of confidence that the input sample belongs to a certain class and say the model has assigned it to a particular class depending on whether the model output exceeds a given threshold or not. The *accuracy* is defined as the percentage of samples which were classified correctly by the model.

A more complex method of evaluation is the receiver operating characteristic (ROC) used in signal processing. It is the dependence of the percentage of samples from class A classified correctly on the percentage of samples from another class incorrectly classified as class A, evaluated as the threshold is varied [297]. While we do not directly apply this metric, it is closely related to the sensitivity curve as described in Sec. 2.4.2 and is applied in the pioneering literature we are following [163].

3.2 Training a model

While the method outlined in Sec. 3.1 seems quite simple, it comes with a large set of caveats and potential complications. Not only is finding the minimum of \mathcal{C} a difficult task by itself, in many cases, the actual global minimum may perform very poorly when presented with new data despite being the best fit to the training data. This section describes some of the main pitfalls and the basic solutions to overcome them.

3.2.1 Optimization

Let us consider the training problem as introduced in Sec. 3.1:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}), \mathbf{Y}) . \quad (3.9)$$

Gradient descent

We first consider a simple iterative process:

$$\mathbf{m}_j = \nabla_{\boldsymbol{\theta}} \mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}_{j-1}), \mathbf{Y}) + \gamma \mathbf{m}_{j-1}, \quad (3.10a)$$

$$\boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1} - \eta \mathbf{m}_j, \quad (3.10b)$$

where the parameters η and γ parameters are named *learning rate* and *momentum*, respectively.

When $\gamma = 0$, this is an optimization method called *gradient descent*, approximating the continuous process

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -\eta \nabla_{\boldsymbol{\theta}} \mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}(t)), \mathbf{Y}), \quad (3.11)$$

which is guaranteed to reduce the loss unless in a point with zero gradients. Eqs. (3.10) are convergent when optimizing a convex function given a suitable value for η , but in a DNN training procedure, the function being optimized often possesses many local minima and saddle points. The gradient descent method struggles with these and can easily get stuck in a local minimum and is therefore unsuitable for this task.

These issues are partly resolved when $\gamma \neq 0$, which is called *gradient descent with momentum*. The corresponding continuous process is described by a second-order differential equation, following the analogy to the inertia of a physical body in motion. This helps the optimizer to traverse local minima. However, there is another issue.

In complex models with many weights, gradients with respect to different weights often differ by several orders of magnitude, and therefore they are optimized much slower. Increasing the learning rate improves the convergence speed of these weights but causes others to diverge, as described further in Fig. 3.1.

Second moment methods

The convergence issue described above can be partly resolved by using second moment methods, which keep a running average of not only the gradient but also its square. Examples include Adagrad [312] and RMSProp [313]. In this thesis, we use an optimizer designed to combine the advantages of Adagrad and RMSProp, the Adam [314]

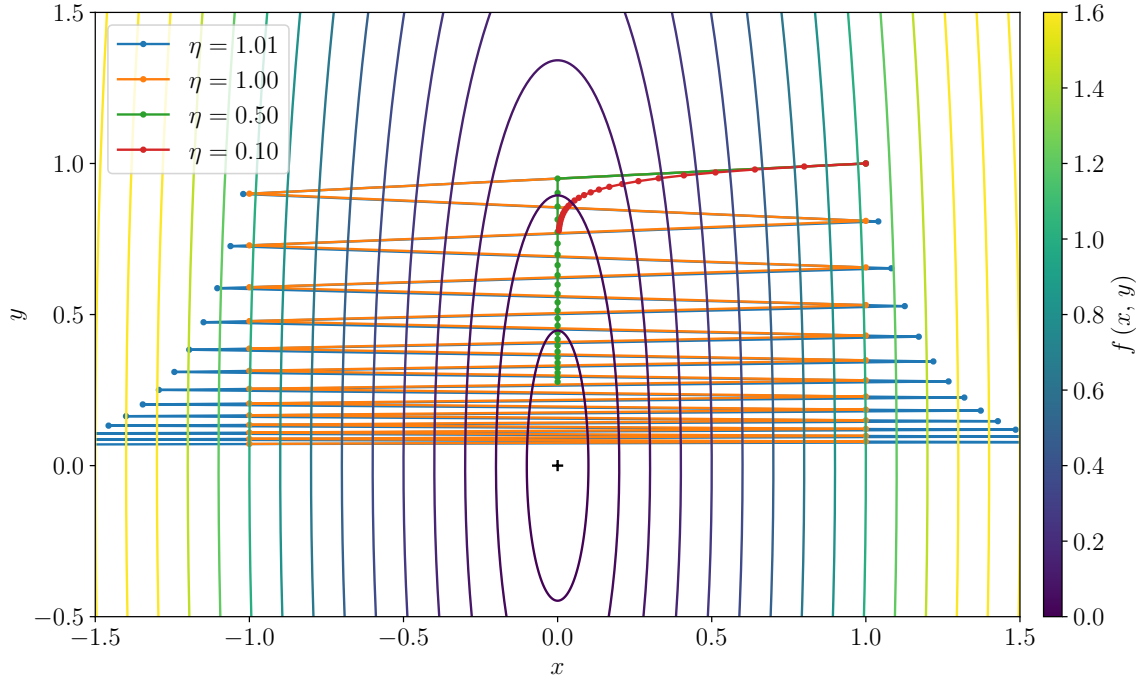


FIGURE 3.1: Gradient descent optimization of a function given as $f(x, y) = x^2 + y^2/20$ using four different learning rates. All optimizations start from the same point $(1, 1)$ and the momentum parameter is set to $\gamma = 0$. The process is separable into the x and y directions and learning rates which converge quickly in one direction either diverge or converge slowly in the other in the highly asymmetric case. A similar effect frequently occurs in landscapes being optimized in neural network training procedures and can be alleviated using second-order gradient-descent based methods.

algorithm, defined as

$$\mathbf{g}_j = \nabla_{\theta} \mathcal{C}(f(\mathbf{X}, \theta_{j-1}), \mathbf{Y}) , \quad (3.12a)$$

$$\mathbf{m}_j = \beta_1 \mathbf{m}_{j-1} + (1 - \beta_1) \mathbf{g}_j , \quad (3.12b)$$

$$\mathbf{v}_j = \beta_2 \mathbf{v}_{j-1} + (1 - \beta_2) \mathbf{g}_j^2 , \quad (3.12c)$$

$$\hat{\mathbf{m}}_j = \mathbf{m}_j / (1 - \beta_1^j) , \quad (3.12d)$$

$$\hat{\mathbf{v}}_j = \mathbf{v}_j / (1 - \beta_2^j) , \quad (3.12e)$$

$$\theta_j = \theta_{j-1} - \eta \frac{\hat{\mathbf{m}}_j}{\sqrt{\hat{\mathbf{v}}_j + \epsilon}} , \quad (3.12f)$$

with parameters:

- η : learning rate
- β_1, β_2 : decay of first and second moment, respectively

- ϵ : regularization parameter

It follows the same basic idea as gradient descent with momentum in Eqs. (3.10). The optimization step in Eq. (3.12f) differs from gradient descent in that the step length in different directions is rescaled by $\left(\sqrt{\hat{\mathbf{v}}} + \epsilon\right)^{-1}$ in order to make the optimization more uniform. This helps to resolve the issue of optimizing a highly anisotropic function. The ϵ parameter is a regularization to prevent division by zero.

The power of this algorithm lies in the reduction of differences in convergence speed in different directions and therefore an increased robustness and speed of optimization. In addition, it is much less sensitive to the choice of learning rate than regular gradient descent; we have found the Adam algorithm to converge in reasonable time for η values up to 6 orders of magnitude apart, and a mere one order of magnitude for the gradient descent method, in searching for the minimum of the Beale function [315, 316].

Stochastic optimization

The methods described above rely on the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}), \mathbf{Y})$ and its repeated computation as the weights $\boldsymbol{\theta}$ are modified with each iteration of the optimizer. In many modern applications, this is computationally too expensive, as the sizes of datasets often go into the millions or higher, and this would lead to infeasible training times. To resolve this, we turn to stochastic methods.

Let us split our dataset (\mathbf{X}, \mathbf{Y}) with m samples into *mini-batches*, also simply called *batches*

$${}^{(j)}\mathbf{X} := (\mathbf{X}_i)_{i=jl}^{(j+1)l}, \quad {}^{(j)}\mathbf{Y} := (\mathbf{Y}_i)_{i=jl}^{(j+1)l}, \quad j = 0, 1, \dots, m/l - 1. \quad (3.13)$$

The parameter l is called the *batch size* and for simplicity, we assume $m/l \in \mathbb{N}$. If the input space \mathcal{X} is sufficiently well sampled and the batch size is high enough, one can approximate the gradient by computing it over individual batches

$$\nabla_{\boldsymbol{\theta}}\mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}), \mathbf{Y}) \approx \nabla_{\boldsymbol{\theta}}\mathcal{C}(f({}^{(j)}\mathbf{X}, \boldsymbol{\theta}), {}^{(j)}\mathbf{Y}), \quad j = 0, 1, \dots, m/l - 1. \quad (3.14)$$

A stochastic optimizer will then iterate over the entire dataset and perform a single optimization step on each batch. Its application to the gradient descent algorithm is well known as stochastic gradient descent (SGD). When using second moment optimizers to train complex neural networks, the word “stochastic” is often omitted but stochastic gradient computation is implicitly used.

Initial state

All methods mentioned in this section are iterative and require an initial set of values for θ . A usual method to acquire the initial θ for the training is random initialization. There are multiple standard distributions from which to draw the initial weights [317–320] for different use cases.

We mention in particular two of the most typical ones: the uniform Glorot initialization [317] and the uniform Kaiming distribution [318], which both consist of a zero-mean non-correlated uniform distribution and an elaborate formula for the boundary, based on the network architecture and the specific layer being initialized. They are used by default by the two most usual neural network frameworks: Kaiming by the PyTorch library, and Glorot by TensorFlow.

In some cases, it has proven useful to use weights obtained through training on a different problem as initial weights for a new training run. This method is called *transfer learning* [321, 322]. In Ch. 4, we test the effects of a method based on transfer learning on the performance of a ML-based GW search.

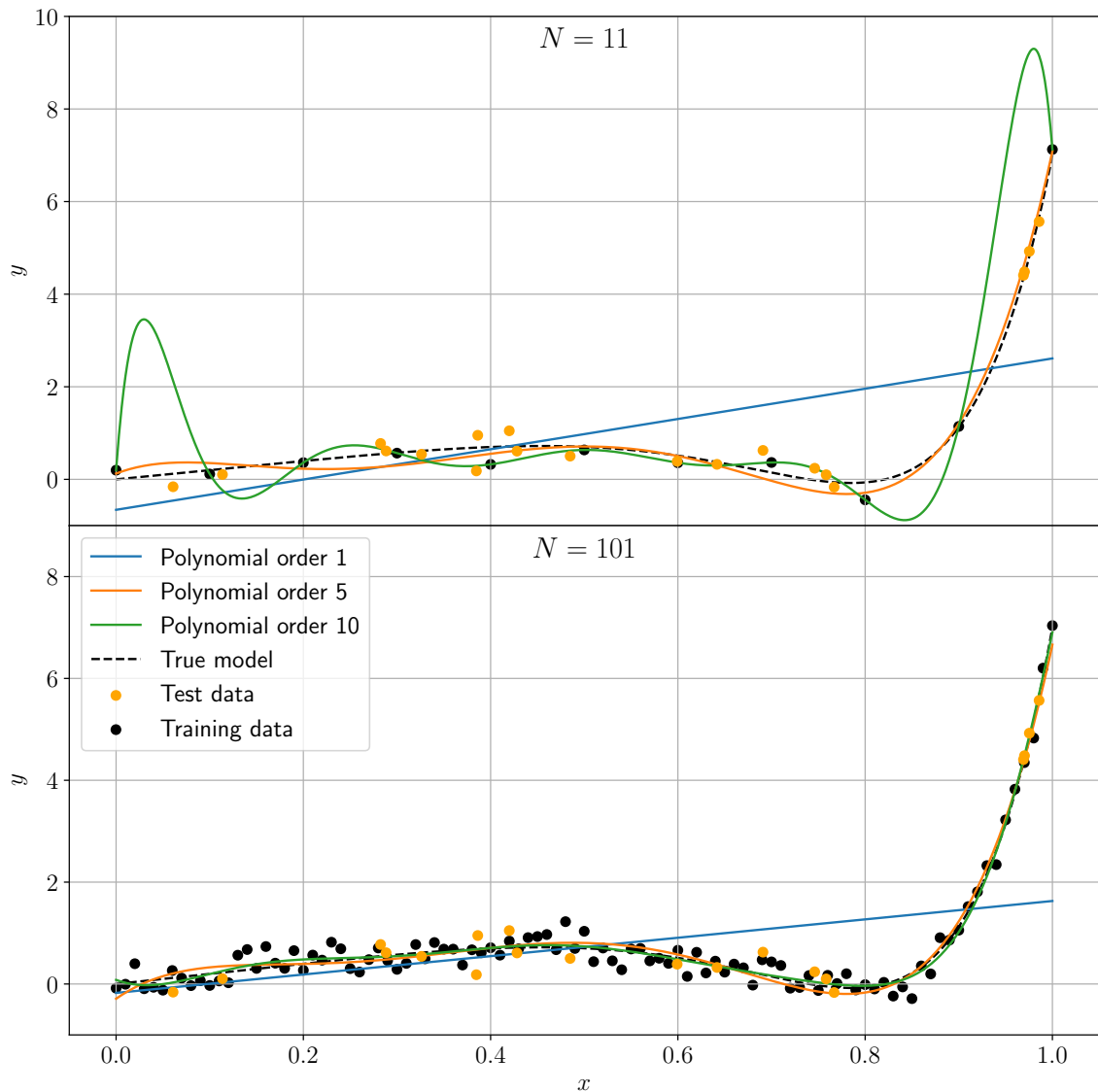
3.2.2 Model selection

As we have mentioned in the introduction of this section, in many cases, the actual global minimum may perform very poorly when presented with new data despite being the best fit to the training data. In the same spirit, when multiple models are trained, the one which performs best on the data used for optimization may prove to be a bad fit on new data, despite being produced by the same distribution. This is a common phenomenon in ML and can be illustrated on a simple example.

Let us use polynomials of various orders to model data which was generated by a noisy polynomial, as shown in Fig. 3.2. The \mathbf{x} values of the training data are 11 and 101 in the top and bottom panels, respectively, uniformly spaced values from 0 to 1. The corresponding \mathbf{y} values are determined by a 10th order noisy polynomial

$$f^*(x) = 15x^{10} - 10x^5 + 2x, \quad \mathbf{y} = f^*(\mathbf{x}) + \boldsymbol{\sigma}, \quad (3.15)$$

where $\boldsymbol{\sigma}$ is uncorrelated Gaussian noise with a standard deviation of 0.2, and operations of the equation above are performed element-wise. We fit polynomials of orders 1, 5 and 10 to the training data to minimize the mean squared error. Considering that the 10th order polynomial is the only model which is in principle capable of reproducing the true model f^* , one might expect it to be the most suitable. Indeed, it does



polynomial order	$N = 11$		$N = 101$	
	training loss	test loss	training loss	test loss
1	2.8	1.8	1.1	2.5
5	$3.8 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$5.0 \cdot 10^{-2}$	$6.9 \cdot 10^{-2}$
10	$5.2 \cdot 10^{-26}$	3.9	$3.7 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$

FIGURE 3.2: Plots of several polynomial fits using different numbers of training points (top panel: $N = 11$, bottom panel: $N = 101$) illustrating the issue of overfitting. The corresponding training and test loss values are in table below, rounded to 2 significant digits. While the 10th order polynomial fits the training data the best in both cases, the 5th order is better at predicting new values not used to train the network in the $N = 11$ case. In the $N = 101$ case, both perform well visually but the 10th order reaches a lower test loss.

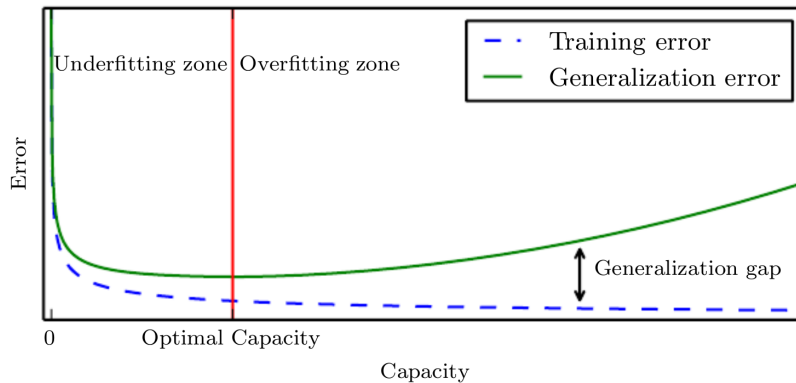


FIGURE 3.3: Simplified illustration of the bias-variance trade-off. The red line marks the ideal model complexity, also called capacity. On the right side, the model is underfitting, i.e. fits the training data better, but fails to generalize to new data. On the left side, the model complexity is insufficient to capture the behavior of the true model, and it is therefore underfitting. Figure taken from [115].

yield the best fit to the training data, being the only model to reproduce it exactly (up to machine precision) in the $N = 11$ case (more specifically, the fit returns the corresponding Lagrange interpolation polynomial [323]).

However, our goal is to predict new data, rather than reproduce the training data. For this reason, one must check the performance of the trained model on independently generated test data. In this experiment, the test data consists of 20 x values drawn from a uniform distribution over the interval $[0, 1]$, and \mathbf{y} computed the same way as in the training dataset (see Eq. (3.15)). In the $N = 11$ case, the 10th order polynomial gives the worst test loss value of the tested models, as can be seen in the table of Fig. 3.2. Clearly, the best approximation of f^* and of the training data is achieved by the 5th order polynomial. In the $N = 101$ case, both higher-order models perform well visually, while the loss values show a lower test loss of the 10th order polynomial. Clearly, a suitable choice of model depends strongly on the number of data points and many other parameters of the problem.

This failure to generalize despite a good fit on a specific dataset is a phenomenon called *overfitting* which may occur in many different applications of ML methods. As seen above, a more complex model may perform worse than a simpler one, depending on multiple factors including the availability of data. At the same time, a model which is too simple will also underperform, which is called *underfitting*. This behavior is part of a more general relationship called the *bias-variance trade-off* [114, 115, 324], further illustrated in Fig. 3.3.

In a typical ML model such as a DNN, however, training using gradient-descent

based methods is much slower and results more difficult to visualize than in our example. In addition, often the model runs into overfitting issues later in the training procedure after passing through some states which perform better, and *early stopping* can be employed [115, 325, 326]. One might try to search for the ideal time to stop the training process by tracking the test loss - however, this indirectly influences the model weights through the test data, which is undesirable and may, in a certain sense, also lead to overfitting.

To resolve this, a third independent set is used, called the *validation dataset*. It never directly influences the training process, the model loss over it is regularly calculated during the optimization and the resulting curve is used to determine at which point the network may have started to overfit and which state is suitable for further testing. Training a complex ML model using a stochastic optimizer then consists of the following steps (or a variation thereof):

1. Perform $E \in \mathbb{N}$ optimization epochs, in each:
 - (a) Loop over training batches, on each:
 - i. Compute batch loss and its gradients
 - ii. Apply the optimizer to tune the weights
 - (b) Compute validation loss
 - (c) Save current network state
2. Select suitable state based on validation losses
3. Calculate the test loss

3.3 Neural networks

In the previous sections, we have introduced the basic procedure of training a ML model to approximate a function using input and output samples. We have not, however, discussed examples of ML models. This section introduces *neural networks*, as a wide class of common ML models. Especially some of their subclasses, such as CNNs, have become very popular in the 2010s due to their great success in many applications, most notably various areas of computer vision [327–332] and associated problems such as autonomous driving [333–336].

3.3.1 The neuron

Let us start by defining a simple function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$f(\mathbf{x}) := \sigma \left(\sum_{i=1}^n w_i x_i + b \right), \quad (3.16)$$

where the linear combination coefficients w_i are called *weights* and the b is called a *bias*, and the σ is an *activation function*. Popular choices include:

$$\text{Identity:} \quad I(z) = z, \quad (3.17a)$$

$$\text{Step function:} \quad \Theta(z) = \begin{cases} 0 & \text{if } z < 0, \\ 1 & \text{if } z \geq 0, \end{cases} \quad (3.17b)$$

$$\text{Sigmoid:} \quad S(z) = \frac{1}{1 + \exp(-z)}, \quad (3.17c)$$

$$\text{Rectified Linear Unit:} \quad \text{ReLU}(z) = \begin{cases} 0 & \text{if } z < 0, \\ z & \text{if } z \geq 0, \end{cases} \quad (3.17d)$$

$$\text{Exponential Linear Unit:} \quad \text{ELU}(z) = \begin{cases} \alpha(\exp(z) - 1) & \text{if } z < 0, \\ z & \text{if } z \geq 0. \end{cases} \quad (3.17e)$$

(typically $\alpha = 1$)

This function f is called an *artificial neuron*. The original idea was using the step function Θ as its activation, and the neuron would then be called a *perceptron*. Its structure is inspired by the function of actual neurons in a human brain: there are multiple inputs from cells in sensory organs, or from other neurons, the neuron performs a simple operation, as a result of which it “fires” ($f(\mathbf{x}) = 1$) or not ($f(\mathbf{x}) = 0$).

However, for training using gradient-descent methods, Θ is unsuitable, as its derivative is zero almost everywhere in \mathbb{R} , which prevents weight optimization. Traditionally, this has driven interest in using the sigmoid S or the closely related hyperbolic tangent, as they approach the same asymptotes as Θ . Lately, however, there has been great interest in using *linear units*, such as ReLU and ELU (see above), due to their quick and simple evaluation as well as numerical properties.

When training through loss minimization, both the weights w_i and the bias b are optimized. Therefore, they are somewhat confusingly both called weights of the model.

3.3.2 The dense neural network

We can then build a basic *neural network* by using m neurons with independent sets of weights to produce an m -dimensional output. The weights can be organized in an $m \times n$ matrix and the biases in a vector of length m

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m , \quad (3.18a)$$

$$\mathbf{x} \mapsto \bar{\mathbf{y}}, \quad \bar{y}_j = \sigma \left(\sum_{i=1}^n W_{ji} x_i + b_j \right), \quad j = 1, 2, \dots, m . \quad (3.18b)$$

For practicality, we define the extension of the activation functions defined for a one-dimensional input in Eqs. (3.17) to vector inputs element-wise, so that we can also write the equation above in a more compact manner as

$$\bar{\mathbf{y}} = f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) . \quad (3.19)$$

With $m > 1$ outputs, a very useful activation function can be the Softmax

$$\text{Softmax} : \mathbb{R}^m \rightarrow \mathbb{R}^m , \quad \mathbf{z} \mapsto \frac{\exp(\mathbf{z})}{\sum_{i=1}^m \exp(z_i)} , \quad (3.20)$$

producing a set of m non-negative outputs which sum up to 1. For this reason, it is often used for classification problems, the outputs being interpreted as probabilities.

In building more complex networks, this function is referred to as a *dense layer* and often, multiple of them are used. However, it can already be a fairly well-performing network: it is quite simple to achieve an accuracy of over 90% on the MNIST dataset [311] using a shallow network with $n = 28^2 = 784$, $m = 10$, with the Softmax activation, a cross entropy loss and a simple SGD optimizer.

Let us now define a deep neural network (DNN). It is, quite simply, a chain of dense layers, defined in Eqs. (3.18). For an input size n and output size m , let us take a sequence of $L + 1$ positive integers $n^{(0)} = n, n^{(1)}, \dots, n^{(L-1)}, n^{(L)} = m$, and build L layers $f^{(1)}, f^{(2)}, \dots, f^{(L)}$

$$\forall i = 1, 2, \dots, L : \quad f^{(i)} : \mathbb{R}^{n^{(i-1)}} \rightarrow \mathbb{R}^{n^{(i)}} , \quad (3.21a)$$

$$f^{(i)}(\mathbf{x}) = \sigma^{(i)}(\mathbf{W}^{(i)}\mathbf{x} + \mathbf{b}^{(i)}) . \quad (3.21b)$$

As indicated by the indices in brackets, individual layers have independent weights, biases, sizes and even activation functions (although these are frequently the same

through the entire network). They are then applied sequentially to the input, forming the final expression for the network

$$f := f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(2)} \circ f^{(1)} . \quad (3.22)$$

When training a DNN, all weight matrices and bias vectors are optimized, so the full set of free parameters is $\boldsymbol{\theta} = (\mathbf{W}^{(i)}, \mathbf{b}^{(i)})_{i=1}^L$. Somewhat confusingly, the term “weights” can refer either just to the $\mathbf{W}^{(i)}$ matrices, or to all free parameters $\boldsymbol{\theta}$ (cf. Sec. 3.1). For the rest of this thesis, it will refer to the specific matrices when discussing individual layers, and to the full set of parameters when talking about a network in general, which is an established convention in the ML community.

In addition, while we defined a dense layer as the combination of a linear map and an activation function, it is conventional in practical applications to separate them and use the term “dense layer” or “linear layer” to refer to the linear map and define the activation function as a standalone layer. From this point forward, we follow this terminology.

3.3.3 Convolutional neural networks

One of the main applications for DNNs is image processing. Fully connected networks as described in Sec. 3.3.2 are, however, inefficient for this purpose, as the number of trainable weights rises too fast as the input size increases. This can be resolved by leaving out less important connections in the network and sharing weights between some neurons in a manner which preserves the relevant connections and symmetries and allows for a more feasible scaling.

The conventional solution for image processing is to employ a 2-dimensional *convolutional neural network* (CNN). In this work we only require the 1D CNN, which is a simpler version used extensively in time-series applications. Unlike the DNNs discussed above, 1D CNNs deal with two-dimensional data. Thus, when dealing with layers specific to CNNs, matrices denoted by bold uppercase letters refer to individual input samples rather than entire datasets.

To build a typical CNN design, we require three new layers:

- Let us assume a 2-dimensional input tensor \mathbf{X} of the shape $C_{\text{in}} \times L$ (the first being the channel dimension), a weight tensor \mathbf{W} of the shape $C_{\text{out}} \times C_{\text{in}} \times K$

and a bias vector \mathbf{b} of length C_{out} . We then define the 1D convolutional layer as

$$\text{Conv1D}(\mathbf{X}) := \left(b_i + \sum_{j=1}^{C_{\text{in}}} \mathbf{W}_{ij} \star \mathbf{X}_j \right)_{i=1}^{C_{\text{out}}}. \quad (3.23)$$

Its number of trainable parameters is much lower than that of a fully connected layer with equivalent input and output sizes. Not only have many connections between neurons been removed, a lot of weights are also shared between connections. In the 2D case, this is done analogically, computing the cross-correlation along two dimensions (the weights \mathbf{W} also possess an additional dimension).

This definition emulates the inner workings of the visual cortex of some animals [337] and possesses translational symmetry, which is desirable for typical image processing tasks, which attempt to detect edges and extract objects which may be located in different parts of the input image. In a similar manner, one would expect a certain degree of translational invariance in a network meant to process one-dimensional signals.

There are additional parameters which affect the behavior and even output size of a standard convolutional layer, namely the stride and dilation. Multiple padding methods are also frequently used and available in most frameworks. In this work, we limit ourselves to the case where stride = dilation = 1 and no padding is applied, which leads to the simplified definition given above. For details on these extensions, see [124, 126].

- Another crucial component of CNNs are downsampling operations. The most typical is the MaxPool1D layer, for an input shape $C_{\text{in}} \times L$ defined by its output

$$\text{MaxPool1D}(\mathbf{X}) := \left(\max \left(\{X_{i,m+K(j-1)}\}_{m=1}^K \right) \right)_{i=1,j=1}^{C_{\text{in}}, \lfloor L/K \rfloor}, \quad (3.24)$$

where $K \in \mathbb{N}$ is the layer's kernel size (fully independent of the convolution kernel size). This corresponds to splitting the input into tiles of length K and only taking the highest element from each tile as the output. In the same way as the convolutional layer, the MaxPool2D layer is applied over two dimensions in the image processing applications and possesses a certain degree of translational and rotational invariance.

- Finally, the *flattening* layer has no trainable parameters. It merely takes a higher-dimensional tensor and rearranges its elements into a lower-dimensional

one, without changing any of its values. Different options for ordering the output tensor elements exist but are irrelevant when followed by a randomly initialized dense layer, which is very often the case.

Taking these components, the basic CNN design consists of

1. several convolutional blocks, each with
 - (a) a convolutional layer,
 - (b) a downsampling operation,
 - (c) an activation function,
2. flattening,
3. a dense neural network.

3.3.4 Back-propagation

In order to train neural networks using gradient-descent based methods, an efficient gradient computation algorithm is required. While a direct differentiation of Eq. (3.22) iteratively applying the chain rule

$$\frac{d}{dx} (f \circ g) (x) = \frac{df}{dg} (g(x)) \cdot \frac{dg}{dx} (x) \quad (3.25)$$

for each individual weight is possible, it is not computationally efficient when applied to a DNN. In this subsection, we demonstrate a more efficient process, which gradually computes gradients, starting from the output layer and moving towards the input, making use of mean products of the previously computed gradients. This process is called *back-propagation* and it is at the heart of the vast majority of DNN applications today [114, 115, 338].

Let us focus on a single evaluation of a dense neural network f as defined in Eqs. (3.21), (3.22), also called a *feed forward*, from an input vector \mathbf{x} to the output $\bar{\mathbf{y}} = f(\mathbf{x})$. For convenience, we denote some mean products of the process, building

the feed-forward algorithm as

$$\mathbf{a}^{(0)} := \mathbf{x} , \quad (3.26a)$$

$$\forall l = 1, 2, \dots, L : \mathbf{z}^{(l)} := \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} , \quad (3.26b)$$

$$\mathbf{a}^{(l)} := \sigma^{(l)}(\mathbf{z}^{(l)}) , \quad (3.26c)$$

$$\bar{\mathbf{y}} = \mathbf{a}^{(L)} . \quad (3.26d)$$

Note that while we define the loss function over a dataset of individual samples, in this section we work with the loss over a single sample for simplicity.

For training the network, we will typically be taking partial derivatives of a loss function $\mathcal{C} = \mathcal{C}(f(\mathbf{x}, \boldsymbol{\theta}), \mathbf{y})$ with respect to the network weights $\boldsymbol{\theta}$. Moving backwards through the network, one can simply compute the derivatives in matrix form as

$$\frac{\partial \mathcal{C}}{\partial \mathbf{a}^{(L)}} = \frac{\partial \mathcal{C}}{\partial \bar{\mathbf{y}}} , \quad (3.27a)$$

$$\forall l = L, L-1, \dots, 1 : \frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{a}^{(l)}} \odot \sigma^{(l)'}(\mathbf{z}^{(l)}) , \quad (3.27b)$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{a}^{(l-1)}} = \left(\frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} \right)^T \mathbf{W}^{(l)} , \quad (3.27c)$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{W}^{(l)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} \otimes \mathbf{a}^{(l-1)} , \quad (3.27d)$$

$$\frac{\partial \mathcal{C}}{\partial \mathbf{b}^{(l)}} = \frac{\partial \mathcal{C}}{\partial \mathbf{z}^{(l)}} . \quad (3.27e)$$

This provides us with an iterative algorithm, which yields the gradients of the loss function with respect to all the trainable weights through Eqs. (3.27d), (3.27e).

In the case of a more complex architecture such as a CNN, the procedure follows the same basic idea: moving from the output to the input layer and reusing information from the previous layer to compute the partial derivatives of the loss function with respect to network weights. DNN frameworks include a very efficient and versatile implementation of the back-propagation algorithm over a full batch of samples [124, 125, 127, 128].

In a convolutional layer as given by Eq. (3.23), carefully computing the partial derivatives yields

$$\frac{\partial \mathcal{C}}{\partial X_{rs}} = \sum_{k=1}^{C_{\text{out}}} \sum_{i=1}^K \frac{\partial \mathcal{C}}{\partial \bar{Y}_{k(s+1-i)}} W_{kri} , \quad (3.28a)$$

$$\frac{\partial \mathcal{C}}{\partial W_{rst}} = \sum_{l=1}^{L+1-K} \frac{\partial \mathcal{C}}{\partial \bar{Y}_{rl}} X_{s(t+l-1)} , \quad (3.28b)$$

$$\frac{\partial \mathcal{C}}{\partial b_r} = \sum_{l=1}^{L+1-K} \frac{\partial \mathcal{C}}{\partial \bar{Y}_{rl}} . \quad (3.28c)$$

3.4 Regularization

In Sec. 3.2 we have covered some issues one may run into when attempting to train a complex model. A set of techniques to counter them summarily called *regularization* have been developed [115, 339] and this section describes several of them.

3.4.1 Exploding gradients

Depending on the training setup, the loss may diverge at specific points and in its vicinity, its gradients may become very large. This breaks the approximate equivalence of Eqs. (3.10) and (3.11) and the optimizer may perform very large steps which deteriorate the network's performance. This can be resolved using *gradient norm clipping* [340, 341], which replaces the gradients by

$$\text{clip}(\nabla_{\boldsymbol{\theta}} \mathcal{C}(f(\mathbf{X}, \boldsymbol{\theta}), \mathbf{Y}), g_{\text{max}}) , \quad (3.29a)$$

$$\text{clip}(\mathbf{g}, g_{\text{max}}) := \begin{cases} (g_{\text{max}} / \|\mathbf{g}\|) \cdot \mathbf{g} & \text{if } \|\mathbf{g}\| \geq g_{\text{max}} , \\ \mathbf{g} & \text{otherwise ,} \end{cases} \quad (3.29b)$$

where g_{max} is a positive parameter, $\|\cdot\|$ is any norm and the clip in the top equation is performed over a single vector of gradients with respect to all trainable parameters. In some cases, it may also accelerate the training procedure [342].

The categorical cross-entropy loss (see Eqs. (3.6)) is divergent when for a certain i, j pair it holds $\bar{Y}_{ij} = 0, Y_{ij} > 0$, which then leads to exploding gradients. To counter

this issue, we have designed a simple regularized loss

$$\text{CCE}_{\text{reg}}(\bar{\mathbf{Y}}, \mathbf{Y}) := -\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \log((1-\varepsilon)\bar{Y}_{ij} + \varepsilon)Y_{ij}, \quad (3.30)$$

where $0 < \varepsilon \ll 1$ is a regularization parameter. This way, the argument of the logarithm is never less than ε , preventing any divergences. Note that the loss is linear with respect to the labels, which is a useful property due to the approximation of Bayesian posteriors as described in Eq. (4.1).

3.4.2 Overfitting

In cases where a complex model is required and the available data is insufficient, one will easily run into issues with overfitting. To resolve this, a set of techniques exists called *data augmentation*, which effectively increase the size of the dataset by deriving new samples based on existing ones [343, 344]. In the case of image data, these typically consist of random cropping, rotations, mirroring, zoom, or noise injection; in case of audio data, time shifting or noise injection; or in the case of text data, replacement of words with synonyms.

We use a related technique, called a *dropout layer* [345, 346]

$$\text{Dropout} : \mathbb{R}^m \rightarrow \mathbb{R}^m, \quad (3.31a)$$

$$\mathbf{x} \mapsto \bar{\mathbf{y}}, \quad (3.31b)$$

$$\bar{y}_j = \begin{cases} 0 & \text{probability } p, \\ \frac{x_j}{1-p} & \text{probability } 1-p, \end{cases} \quad (3.31c)$$

where $p \in (0, 1)$ is a parameter. Typically, this dropout layer is only inserted during training but removed for evaluation (in fact, for simplicity, PyTorch allows to keep the layer and offers the model to switch between a training mode, where the Dropout behaves as above, and an evaluation mode, where it acts as an identity function).

In a CNN, it is usually inserted in the later part, consisting mainly of dense layers (variants exist, however, which zero out an entire channel in the convolutional part [124, 347]). While not a noise injection per se, it does introduce additional stochasticity and the effect is similar to that of a noise injection method.

Chapter 4

Single-detector searches and training strategies

The first steps into GW data analysis using ML-based algorithms have been done in [162, 163], where CNNs following the basic design shown in Sec. 3.3 were trained using generated data to distinguish samples consisting of detector noise and a non-spinning BBH waveform injection from those only containing pure noise, and evaluated using traditional ML metrics such as the accuracy and the ROC.

One of the questions raised by the two publications is the choice of SNRs of waveform injections in the training dataset. In [162], a method called *curriculum learning* was applied. It consists of starting the training with very loud signals and gradually reducing their SNRs during the training and it is a form of transfer learning [348–350]. The two initial publications, however, disagree on its effect: while [162] found a noticeable improvement in its use, [163] found no difference. This is one of the issues addressed in this project.

Furthermore, the pioneering papers succeeded in classifying isolated data samples at false alarm probabilities (FAPs) down to 10^{-3} (per noise sample analyzed). To obtain a GW search algorithm, it is necessary to extend these results from isolated data samples to continuous streams of data and evaluate them using task-appropriate metrics in use by the GW data analysis community (see Sec. 2.4.2). Namely, these are the FAR and the sensitive distance [52]. In order to compete with traditional GW search algorithms, this also includes a reduction of the FAP by several orders of magnitude.

The contents of this chapter describe a project conducted by a collaboration of the Albert-Einstein-Institut Hannover (represented by Marlin Schäfer, Alex Nitz and Frank Ohme) and the FSU Jena (represented by myself and prof. Bernd Brügmann), which reproduced the results described in [163] and extended them to achieve the

goals above, and culminated in the publication of [215]. All relevant code and data are available publicly as part of the data release at [351].

This work has also been successfully extended to a two-detector search by applying the network to both detector streams and using a simple probabilistic formula to search for coincident triggers [352].

4.1 Experimental setup

Following [163], the training data consists of 1 second samples of Gaussian noise and non-spinning BBH injections. The waveforms are generated using the model `SEOBNRv4_opt` [353], which is an optimized version of the EOB model `SEOBNRv4` [252], through `PyCBC` [354] and, by extension, `LALSuite` [355, 356]. As we are working with a single detector and the waveforms only include the dominant modes $(2, \pm 2)$, the full waveform morphology can be obtained by varying the component masses m_1, m_2 , the coalescence phase Φ_0 , and either the luminosity distance or the optimal SNR, fixing all other parameters.

We choose to set the sky position to be overhead the LIGO Hanford detector [357] and the inclination as well as the polarization to 0. All generated waveforms are rescaled to obtain a given optimal SNR, therefore the choice of distance is irrelevant ($d_L = 1$ Mpc is passed to the approximant). Pairs of masses are drawn from a uniform distribution over $[10M_\odot, 50M_\odot]$ and the higher and lower value is assigned to m_1 and m_2 , respectively, to satisfy the definition of the primary and secondary masses. For each mass pair, 5 different coalescence phases are drawn uniformly from $[0, 2\pi)$, and the resulting waveforms are combined with 5 independently generated noise samples.

While [163] uses a sampling rate of 8192 Hz, all BBH signals from the chosen mass distribution merge at lower frequencies than 1024 Hz and the detector sensitivity drops sharply at both high and low frequencies. For this reason, we restrict our search to the frequency range $[20 \text{ Hz}, 1024 \text{ Hz}]$. This is a standard procedure done in current searches as well and we have verified that little to no SNR is lost due to this. In accordance, we reduce the sampling rate to 2048 Hz for computational efficiency while reaching the Nyquist frequency of 1024 Hz [358]. As our networks converge to results comparable to [163], we are confident that this reduction has no negative impact on the resulting network's performance.

The waveform approximant lower-frequency cutoff is set to 20 Hz. The signal is then projected onto the Hanford detector, whitened by the `aLIGOZeroDetHighPower`

PSD [355, 359, 360], and its optimal SNR is scaled to 1. To reduce the trained network’s sensitivity to the exact alignment of the waveform in the input window, the merger is placed randomly in the [0.6 s, 0.8 s] range within the 1 second-long window.

All noise is generated by the same `aLIGOZeroDetHighPower` PSD used to whiten the waveforms and whitened by the same procedure. Whitening and storing them separately allows us to dynamically change the optimal SNRs of injections during the training procedure, which is desirable as we test training strategies (see Sec. 4.3).

In this way, the training dataset is generated. It consists of 200 000 independent noise samples and 100 000 waveforms stemming from 20 000 independent component mass pairs. When combined, 100 000 noise samples are used with a BBH injection and 100 000 without one, forming the “signal” and “noise” classes. The validation dataset is assembled in the same way and has the same numbers of samples as the training dataset, as it may directly influence the training process through training strategies.

Due to this, metrics derived off of the validation dataset may be biased and we require a third dataset called the *efficiency set*, whose usage is described below. It consists of 400 000 independent noise samples and 10 000 waveforms and is not assembled the same way as the training and validation sets.

Ideally, one would train the network to approximate the Bayesian posteriors $\mathbf{y} = (p(\mathcal{H}_1|\mathbf{d}), p(\mathcal{H}_0|\mathbf{d}))$. Following [189], taking the cross entropy (see Eq. (3.6)) over a given prior distribution yields

$$\begin{aligned}
\langle \mathcal{C} \rangle &= - \int p(\mathbf{d}) \, \mathrm{d}\mathbf{d} [p(\mathcal{H}_1|\mathbf{d}) \log f(\mathbf{d})_0 + p(\mathcal{H}_0|\mathbf{d}) \log f(\mathbf{d})_1] \\
&= -p(\mathcal{H}_1) \int p(\mathbf{d}|\mathcal{H}_1) \, \mathrm{d}\mathbf{d} \log f(\mathbf{d})_0 - p(\mathcal{H}_0) \int p(\mathbf{d}|\mathcal{H}_0) \, \mathrm{d}\mathbf{d} \log f(\mathbf{d})_j \\
&\approx -\frac{p(\mathcal{H}_1)}{N_{\text{wave}}} \sum_{j=1}^{N_{\text{wave}}} \log f(X_j^{\text{wave}})_0 - \frac{p(\mathcal{H}_0)}{N_{\text{noise}}} \sum_{j=1}^{N_{\text{noise}}} \log f(X_j^{\text{noise}})_1 \\
&= -\frac{1}{N} \sum_{j=1}^N [\log(\bar{Y}_{j0}) Y_{j0} + \log(\bar{Y}_{j1}) Y_{j1}] \quad , \tag{4.1}
\end{aligned}$$

where f is the model under evaluation and \mathbf{X}^{wave} and $\mathbf{X}^{\text{noise}}$ are datasets containing and not containing injections, respectively. These have lengths N_{wave} and N_{noise} and follow their corresponding distributions $p(\mathbf{d}|\mathcal{H}_1)$ and $p(\mathbf{d}|\mathcal{H}_0)$, respectively. Naturally, it holds $N = N_{\text{wave}} + N_{\text{noise}}$ and the priors are $p(\mathcal{H}_1) = N_{\text{wave}}/N$, $p(\mathcal{H}_0) = N_{\text{noise}}/N$. Finally, the labels \mathbf{Y}_j are set to (1, 0) for a sample with an injection, and (0, 1) for a pure noise sample. We call the network output probability \bar{Y}_{j0} the *p-score*.

This means that the BCE loss measured with respect to the Bayesian posteriors as labels can be approximated using the so-called *one-hot* labels as defined above. This approximation only holds over a sufficiently large dataset to justify the approximation of the average by a sum over samples following the corresponding distribution as in Eq. (4.1).

Therefore we use one-hot labels in our training. Due to exploding gradients during training, instead of the BCE, the regularization of Eq. (3.30) is used

$$\text{BCE}_{\text{reg}}(\bar{\mathbf{Y}}, \mathbf{Y}) = -\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \log((1-\varepsilon)\bar{Y}_{ij} + \varepsilon)Y_{ij}, \quad (4.2)$$

which converges to the BCE as $\varepsilon \rightarrow 0$. We set $\varepsilon = 10^{-6}$. It should be noted that the critical property for the derivation in Eq. (4.1) is that the loss is a linear function of the label probability, and this property is preserved in the regularized loss. We make use of the Adam optimizer (see Eqs. (3.12)) with the learning rate set to $\eta = 10^{-5}$ and the others at their default values $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ [314].

As the network is trained, its performance is evaluated on the efficiency set. Traditionally, one would use metrics such as accuracy, precision and recall [297], or the ROC curve, such as in [163]. To optimize for search performance, we develop a metric closely related to the ROC curve as well as to those discussed in Sec. 2.4.2.

We define the *false alarm probability (FAP)* of the network at a fixed threshold as the fraction of noise samples which exceed this threshold when the network is applied. We define the efficiency as the true-positive probability at a fixed FAP. To compute it, we sort the p-score outputs of the network over all noise samples in the efficiency set, whose number we denote N_n . Then we compute

$$x = \lfloor N_n \cdot \text{FAP} \rfloor, \quad (4.3)$$

where $\lfloor \cdot \rfloor$ denotes the flooring operation, and choose the x -th largest of the sorted p-scores as the detection threshold. Then, the waveforms are injected into noise samples from the efficiency set at a fixed SNR and evaluated by the network. We calculate the percentage of those which exceed the detection threshold and call it the *efficiency* of the network. In this chapter, the $\text{FAP} = 10^{-4}$ value is fixed for all efficiency calculations, and SNRs 3, 6, 9, 12, 15, 18, 21, 24, 27 and 30 are used. Where a single efficiency is given, it is an arithmetic mean of the efficiencies at these SNRs.

layer type	kernel size	output shape
Input + BatchNorm1D		1×2048
Conv1D + ELU	64	8×1985
Conv1D	32	8×1954
MaxPool1D + ELU	4	8×488
Conv1D + ELU	32	16×457
Conv1D	16	16×442
MaxPool1D + ELU	3	16×147
Conv1D + ELU	16	32×132
Conv1D	16	32×117
MaxPool1D + ELU	2	32×58
Flatten		1856
Dense + Dropout + ELU		64
Dense + Dropout + ELU		64
Dense + Softmax		2

TABLE 4.1: The CNN architecture used for training. Note that the shapes are given as used by PyTorch. When using TensorFlow or other packages which put the channel dimension after the spatial one, the two-dimensional tensors before the Flattening layer are transposed.

The CNN architecture used for this project is based on [163] with several modifications. Primarily due to the input size reduction from 8192 to 2048, the MaxPool1D sizes have been reduced and the input size of the first dense layer after flattening has been changed accordingly. Furthermore, due to initial issues with training, we have decided to add a batch normalization layer [361] in the beginning of the network, known to accelerate training. The final architecture used for training is summarized in Tab. 4.1.

The network was implemented twice and trained using both implementations. One has been performed using the Python framework PyTorch [124] by myself, while the other uses the package TensorFlow [125, 126] and has been done by Marlin Schäfer. We have thus verified that different frameworks provide equivalent results.

4.2 Unbounded Softmax replacement

The left panel of Fig. 4.1 shows the efficiency of a network trained with the optimal SNRs drawn from the [5, 15] interval (the “fixed low mean”, see Sec. 4.3) as a function of the optimal SNR of the waveforms in the efficiency set. While it reproduces the results of [163] at FAPs down to 10^{-3} , reducing it to 10^{-4} causes the network performance to degrade to zero efficiency.

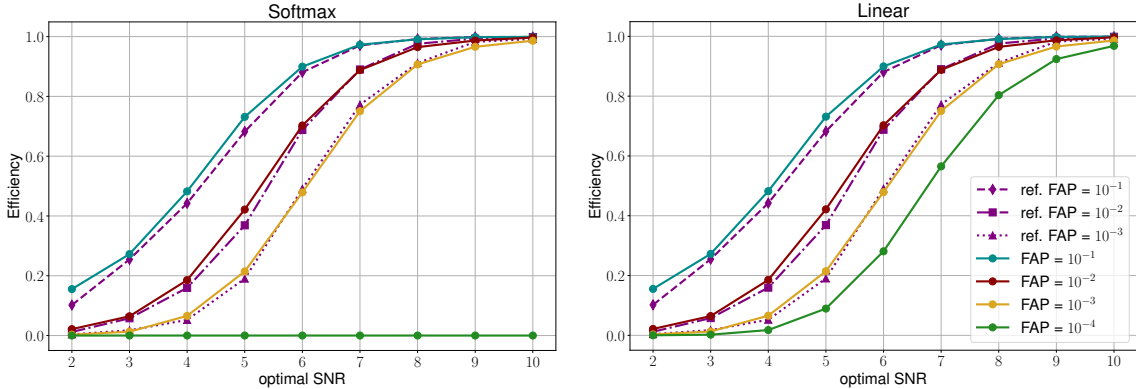


FIGURE 4.1: The efficiency of the “fixed low mean” network evaluated at different optimal SNRs and FARs. In the left panel, the network is used as it was trained, with a Softmax as its final layer. The lines labeled “ref.” are the results of Fig. 3 in [163]. In the right panel, results of the same network are shown, evaluated using the USR modification.

Further analysis reveals that out of the 400 000 samples used to fix $\text{FAP} = 10^{-4}$, over 40 are classified by the network at a p-score of 1. Therefore, during the efficiency calculation, the trigger threshold is set to 1, which is the highest value that the network can output and none of the samples containing waveform injections can exceed it, resulting in an efficiency of 0. This is due to a rounding error in the Softmax layer. Below, we develop a simple modification which removes this issue.

Let us denote by x_0, x_1 the outputs of the last dense layer in the network, i.e. inputs to the Softmax layer. Its first output (the p-score) can be expressed in exact arithmetic as

$$\text{Softmax}(x_0, x_1)_0 = \frac{\exp(x_0)}{\exp(x_0) + \exp(x_1)} = \frac{1}{1 + \exp[-(x_0 - x_1)]}. \quad (4.4)$$

In fact, ML frameworks employ this form when $x_0 > x_1$ for numerical stability, as its definition is much more prone to overflows. While the Softmax output is less than one in exact arithmetic, for $x_0 - x_1 \gtrsim 16$ it is rounded up to 1 in single-precision arithmetic. While raising the floating-point precision would be helpful, even in quadruple precision this issue will occur at $x_0 - x_1 \gtrsim 45$, which is a value reached by some noise samples in our experiment.

To resolve this, we turn to a different ranking statistic with better numerical properties. One can choose any purely growing function of the p-score. We propose two

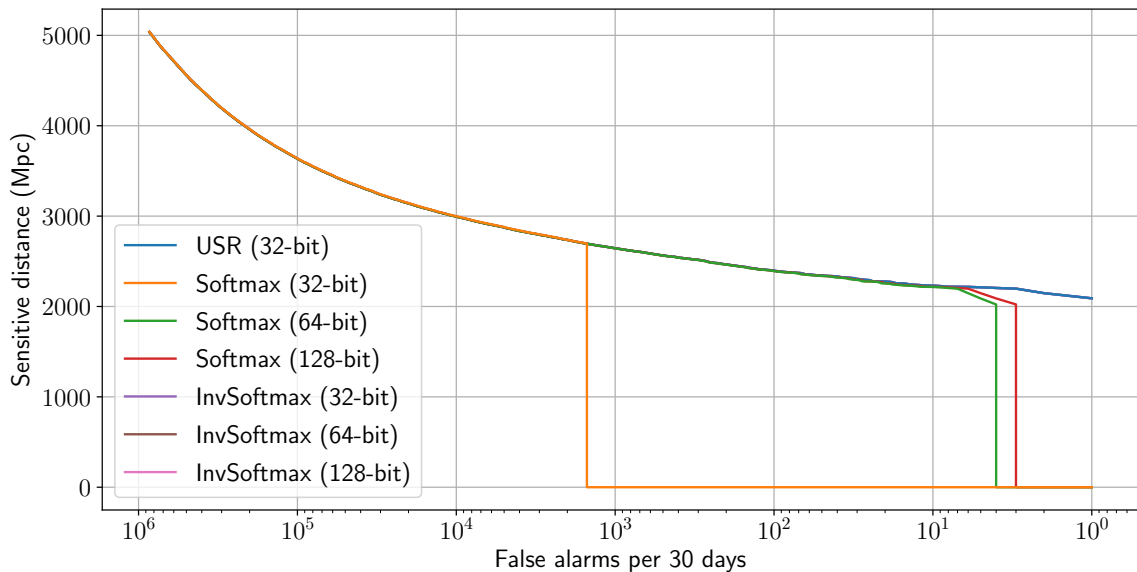


FIGURE 4.2: Comparison of sensitivity curves of the “fixed low mean” model with the three different choices of the final activation evaluated at different floating-point precisions where applicable. All three InvSoftmax curves as well as the USR 32 curve overlap through the whole range and provide comparable performance. However, the original Softmax activation degrades in performance with respect to the USR even when quadruple-precision arithmetic is used. For details on the evaluation of the sensitivity curves, see Sec. 4.4.

suitable choices:

$$\text{USR}(x_0, x_1) := x_0 - x_1, \quad (4.5a)$$

$$\text{InvSoftmax}(x_0, x_1) := -\text{Softmax}(x_0, x_1)_1 = -\frac{1}{1 + \exp(x_0 - x_1)}. \quad (4.5b)$$

As the p-score is a purely growing function of $x_0 - x_1$, we can freely choose the former as our ranking statistic instead. Since this maps the original output range from $(0, 1)$ to \mathbb{R} , we call this modification the unbounded softmax replacement (USR).

The InvSoftmax is also a purely growing function of the p-score:

$$\text{InvSoftmax}(x_0, x_1) = \text{Softmax}(x_0, x_1)_0 - 1, \quad (4.6)$$

therefore it is a valid ranking statistic. It pushes the problem described above to $x_0 - x_1 \gtrsim 104$ in single precision, $x_0 - x_1 \gtrsim 745$ in double precision, and $x_0 - x_1 \gtrsim 11400$ in quadruple precision floating-point arithmetic. In the denominator, the term vanishing through rounding errors is the constant rather than the exponential, and thus the ordering of values is partly preserved.

On the other hand, the USR removes this issue altogether. Fig. 4.2 shows a set of sensitivity curves of one of the networks, using the original network output and both modifications in different degrees of precision. While even the single-precision InvSoftmax seems to perform comparably to the USR, for robustness and simplicity we have decided to use the USR for evaluation. This leads to the efficiency curves shown in the right panel of Fig. 4.1, resolving the issue. However, it should be noted that based on the rounding thresholds above, the InvSoftmax can also be a suitable option when 64-bit floating point number representation is used.

4.3 Training strategies

We performed a set of experiments to investigate the effect of curriculum learning. We have designed 15 ways of adjusting the training SNRs throughout the training, which we call *training strategies*, and they are summarized in Tab. 4.2. They are split into two categories called “curriculum” and “fixed”. The latter consists of 5 strategies, in 3 of which the optimal SNR of the waveforms in the training dataset is a single fixed value (30, 15, and 8, respectively), and in the remaining two an interval is chosen and the optimal SNR for each training sample is drawn randomly from a uniform distribution over the interval. This interval is $[5, 15]$ for the “low” strategy and $[5, 100]$ for the “full” strategy.

The curriculum category consists of 10 strategies which start by drawing the optimal SNRs randomly from the $[90, 100]$ interval and gradually change it to lower values. We chose 5 conditions which determine when to reduce the SNR range and 2 methods by which the new range is determined. In the strategies not marked “relative”, both ends of the interval are reduced by 5 each time the relevant condition is fulfilled until $[5, 15]$, which then remains unchanged for the remainder of the training procedure.

In the strategies marked “relative”, both ends of the range are reduced by 10% of the previous value until the lower end falls below 5. This also reduces the length of the interval by 10% in each step. Once the final step brings its lower end below 5, the entire interval is shifted up to be on this threshold. The final interval is then approximately $[5, 5.523]$.

In the “epochs” condition, this step is performed once every 5 training epochs, while the others are based on metrics computed over the validation dataset. After every epoch, the loss and accuracy over the validation dataset are computed. In the “accuracy” strategy, the condition is the validation accuracy reaching 0.95, and in

Type	Name	Condition
Curriculum	accuracy	when validation
	accuracy relative	accuracy ≥ 0.95
	epochs	every 5 epochs
	epochs relative	
	loss	when validation
	loss relative	loss ≤ 0.2
	plateau accuracy	6 epochs validation
	plateau accuracy relative	accuracy plateau
	plateau loss	6 epochs validation
	plateau loss relative	loss plateau
Fixed	30	SNR = 30
	15	SNR = 15
	8	SNR = 8
	low	SNR $\in [5, 15]$
	full	SNR $\in [5, 100]$

TABLE 4.2: Summary of training strategies used. Curriculum strategies not marked “relative” reduce both ends of the interval by 5 whenever the condition in the last column is fulfilled, while those marked “relative” instead reduce both ends by 5% of the previous value. This is performed until the lower end reaches 5. In the “fixed” strategies, the SNR values are drawn from a uniform distribution over the interval in the last column, or set to a single value where specified.

the “loss” strategy it is the validation loss dropping to 0.2. In the strategies marked “plateau”, the condition is that the corresponding metric has failed to improve by more than 0.01% for 6 consecutive epochs.

To eliminate the bias of favorable weight initializations, the network is trained 50 times using each strategy, resulting in a total of 750 training runs. Three runs using each strategy have been selected for further evaluation using the efficiency values. First, the average efficiency over all 10 SNR values 3, 6, . . . , 30 is computed at each epoch of each run. The epoch is selected as the maximum of this efficiency averaged over all 50 runs of the corresponding strategy. Finally, three network states are chosen at this epoch based on the average efficiencies of the runs. The “high” and “low” refer to the states with the highest and lowest efficiency, and “mean” the closest to average. Over all 15 strategies, this leads to a total of 45 network states being evaluated in the following section.

Figs. 4.3, 4.4 show the efficiency evolution over all 750 training runs, evaluated at SNR = 9. While we monitor other SNRs as well, we are particularly interested in this region for three reasons. First, at SNR ≥ 12 , almost all networks recover close

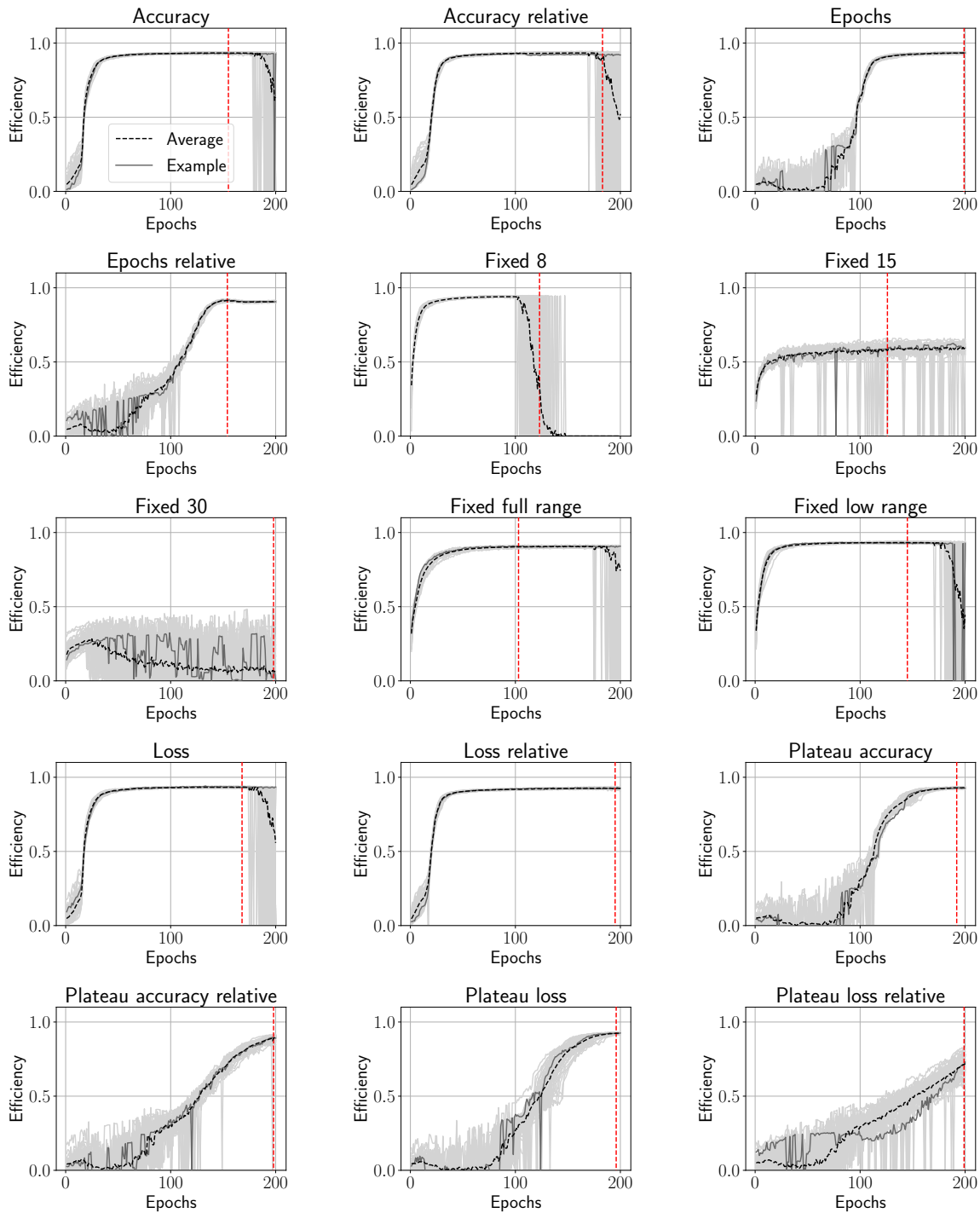


FIGURE 4.3: Efficiencies of all training runs by training strategy, evaluated at $\text{SNR} = 9$, $\text{FAP} = 10^{-4}$. Each panel shows 50 efficiency curves corresponding to all runs using a particular training strategy in light gray, and one randomly chosen run highlighted in dark gray. The dashed black line shows their average and the dashed red line marks the maximum of the efficiency averaged over all SNRs as well as over all fifty runs of the given strategy, which is chosen to select the three states meant for final evaluation. All efficiencies are computed using the Softmax activation.

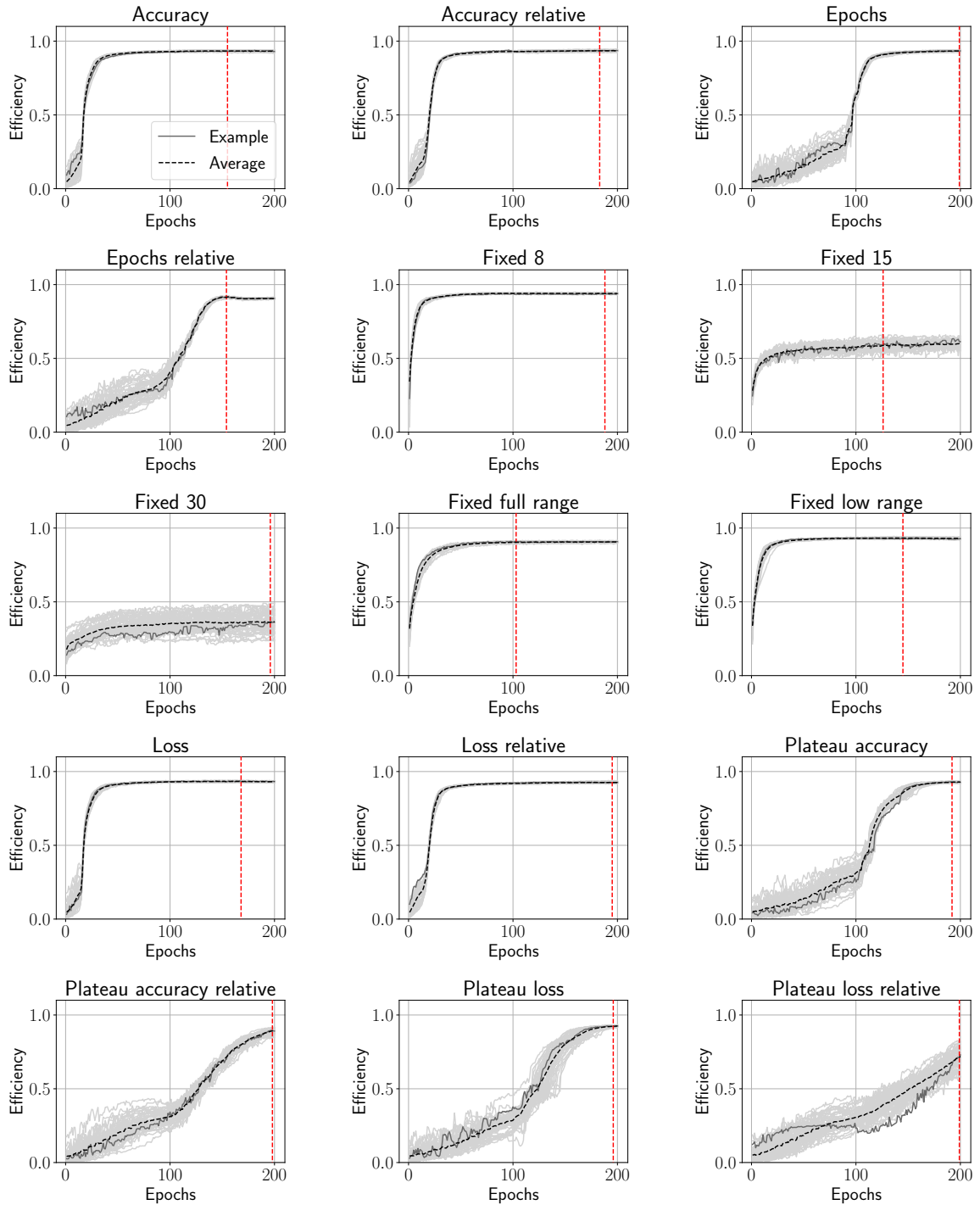


FIGURE 4.4: Efficiencies of all training runs by training strategy, evaluated at $\text{SNR} = 9$, $\text{FAP} = 10^{-4}$. Each panel shows 50 efficiency curves corresponding to all runs using a particular training strategy in light gray, and one randomly chosen run highlighted in dark gray. The dashed black line shows their average and the dashed red line marks the maximum of the efficiency averaged over all SNRs as well as over all fifty runs of the given strategy, which is chosen to select the three states meant for final evaluation. All efficiencies are computed using the USR modification.

to 100% of signals and cannot be separated by efficiency. Second, most GWs are expected to be found at low SNRs, which makes these values most important. Third, $\text{SNR} = 8$ is often used as a threshold above which matched-filter searches can safely detect most signals, and so probing near this value is a very good overall performance measure. More detailed plots at selected SNRs are available for all strategies in the data release [351].

Fig. 4.3 contains drops to zero efficiency on many individual runs and in the case of some strategies, of all the given runs. Further analysis reveals that the trigger threshold in these cases is equal to 1, and is therefore a consequence of the issue described in Sec. 4.2. Fig. 4.4 shows the efficiency evolution of the same runs as Fig. 4.3 with the USR applied. All efficiency drops observed in Fig. 4.3 disappear when the USR is applied.

Let us further examine Fig. 4.4 to determine the effect of training strategies on the trained network efficiencies. Most training strategies do not have a major impact, as all converged networks except “fixed 15” and “fixed 30” reach efficiencies from 90% to 94%. At $\text{SNR} = 6$ the efficiency consistently drops between 24% and 33%, and at $\text{SNR} \geq 12$ all strategies besides the two exceptions reach 100%. The “fixed 15” strategy reaches this milestone at $\text{SNR} = 15$ and “fixed 30” only at $\text{SNR} = 21$.

Efficiencies of the “fixed low” and “fixed 8” runs at $\text{SNR} \geq 18$ have managed to match those of other runs, despite only seeing weaker signals in their training. Conversely, “fixed 15” and “fixed 30” perform well at high SNRs but much worse than others at $\text{SNR} \leq 12$. This implies that CNNs can learn to generalize from weaker signals to stronger ones but not vice versa.

The relative plateau strategies did not converge within the 200 epochs. An additional 200 were performed and this has allowed them to converge. In the end, all converged runs except “fixed 15” and “fixed 30” reached very similar efficiencies and differ mainly in the number of epochs it took to reach their optimal states. We therefore conclude that the choice of training strategy is of little importance to the eventual performance of the network. Instead, it is crucial to include low SNR signals in the training dataset to achieve optimal performance.

In addition, we have repeated these experiments while varying the network hyperparameters, to ensure the robustness of our results to small differences in the network architecture. We found no significant differences in the final efficiencies, merely variations in the training time.

Parameter	Uniform distribution
Approximant	SEOBNRv4_opt
Component masses	$m_1, m_2 \in [10M_\odot, 50M_\odot]$
Spins	$\chi_1, \chi_2 = 0$
Coalescence phase	$\Phi_0 \in [0, 2\pi)$
Polarization	$\Psi \in [0, 2\pi)$
Inclination	$\cos \iota \in [-1, 1]$
Declination	$\sin \theta \in [-1, 1]$
Right ascension	$\varphi \in [-\pi, \pi)$
Luminosity distance	$d_L^2 \in [500^2 \text{ Mpc}^2, 7000^2 \text{ Mpc}^2]$

TABLE 4.3: Injection parameters for the test dataset. A uniform distribution is applied over the intervals in the second column. The injection times are determined by drawing the time between any two consecutive injections from a uniform distribution over [16 s, 22 s].

4.4 Search evaluation

As described in Sec. 2.4.2, a traditional GW search pipeline is expected to produce a set of candidate events, each accompanied by a ranking statistic. To build a search algorithm based on our CNN, we employ the *sliding window* approach. The test data is first sliced into 1 s segments at a step size of 0.1 s to ensure that in any potential waveform, the merger time is always in the same interval as in the training data in at least one slice. These slices are then whitened and fed to the network and the outputs which exceed a given trigger threshold are clustered by time. The maximal separation of two successive triggers in a cluster is 0.2 s. Each cluster is then a candidate event in the output, the time associated with a given event is the first time when the maximum p-score is reached, and the corresponding ranking statistic is that of the maximum.

To evaluate the sensitivity of this algorithm, a test dataset has been generated. It consists of 1 month of simulated Gaussian noise generated by the same PSD as the training data with injections separated by a time randomly drawn from a uniform distribution over the interval [16 s, 22 s]. The injections are generated by the same SEOBNRv4_opt approximant as the ones used for training. The injection parameters are summarized in Tab. 4.3.

As a baseline, we have run the PyCBC [51–55] matched-filter search. The corresponding template bank covers the component mass range $[10M_\odot, 50M_\odot]$. Due to the discretization of the mass space, the maximization in Eq. (2.45b) is inexact. The distribution of masses in the template bank is performed in a way which guarantees that the loss in SNR due to discretization is $\leq 3\%$. The search is implemented by `pycbc_inspiral` and configured to output a set of times where the convolution of any

template with the data returns $\rho_{\text{MF}} > 5$. The configuration files are included in the data release [351].

All sensitivity curves are evaluated by the procedure described in Sec. 2.4.2 using both the Softmax activation and the USR in case of the CNNs. The trigger threshold was set to 0.1 when using the Softmax, and following Eqs. (4.4), (4.5a), it holds

$$\text{Softmax}(x_0, x_1)_0 = 0.1 \Rightarrow \text{USR}(x_0, x_1) \doteq -2.2, \quad (4.7)$$

therefore we use the threshold -2.2 when the USR is applied. A trigger is marked as a true positive if there was an injection within 0.3 s of the reported event time, otherwise it is considered a false positive.

The resulting sensitivity curves of the networks trained by the “fixed low” strategy are shown in Fig. 4.5. All searches perform comparably at $\mathcal{F} \geq 10^5$ per month. At lower values, the searches using the USR perform slightly worse than the matched-filter search but retain at least 91.5% of its sensitivity down to $\mathcal{F} = 1 \text{ month}^{-1}$. However, the searches using the Softmax activation suffer from the rounding issue at $\mathcal{F} \lesssim 10^3 \text{ month}^{-1}$, essentially losing all sensitivity. Sensitivity curves of all training strategies are available in the data release [351] and they show very similar behavior except for the “plateau accuracy relative” and “plateau loss relative”, which did not fully converge, and “fixed 15” and “fixed 30”, which did not learn to detect signals at $\text{SNR} < 15$ as they were not represented in the training data.

All results above used the exact same PSD for data generation and whitening. However, this is unrealistic, as the PSD in detectors drifts over time [72]. To ensure that the network is robust with respect to small changes to the noise characteristic, we have evaluated the sensitivity on data with the same PSD rescaled by a factor of 1.05. This reduced the sensitive distance at all FARs by $\approx 1/\sqrt{1.05}$, in agreement with the theoretical explanation. To test a more realistic variation, we used 20 PSDs derived from the O3a observing run [362] and chose one as reference. We took ratios of the others and the reference and chose the one with the largest mean deviation from unity, and multiplied it with the PSD used to generate the training data to obtain a realistically varied PSD. Generating and whitening the data by this varied PSD reduced the sensitivity at $\mathcal{F} < 100 \text{ month}^{-1}$ to a similar level as observed for the scaled PSD.

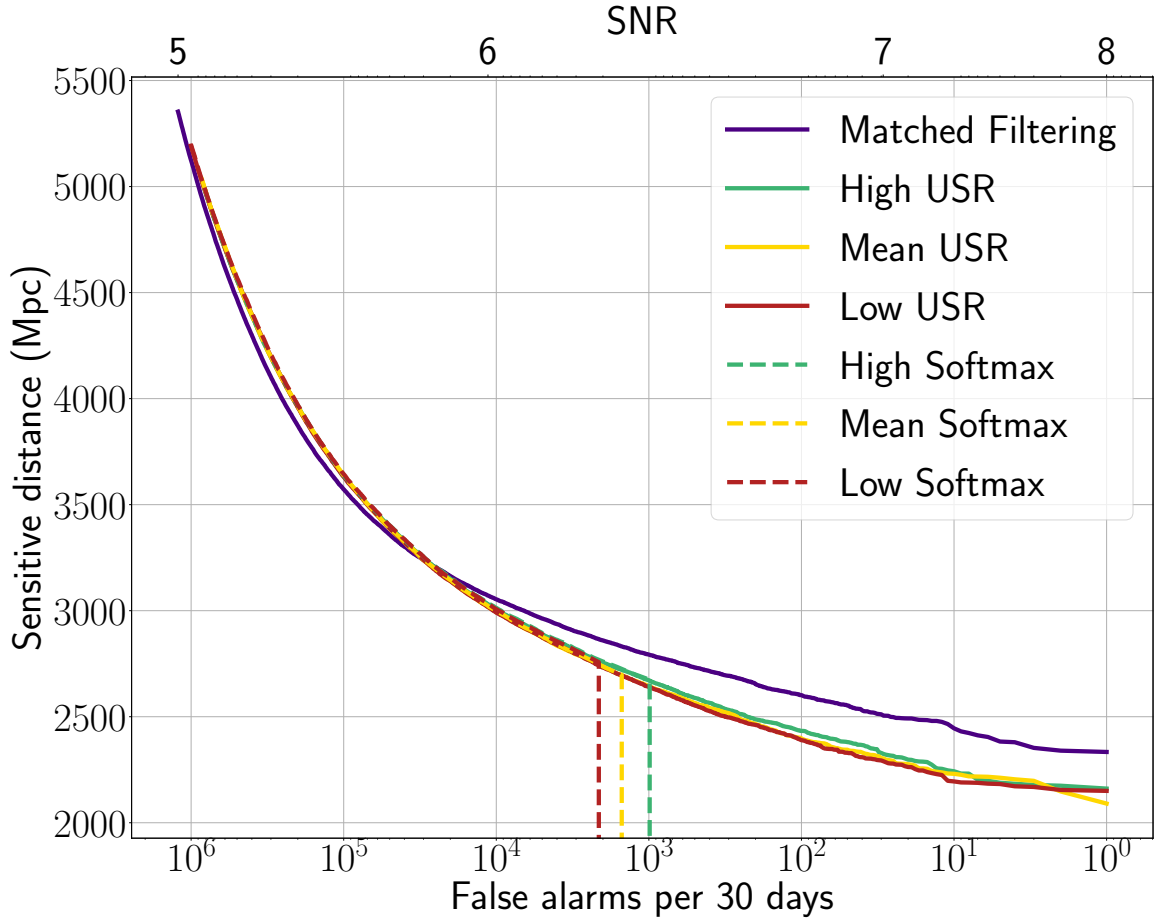


FIGURE 4.5: Sensitivity curves of networks trained by the “fixed low” strategy at epoch 186. The “high”, “mean” and “low” labels refer to the three states selected for further analysis. The curve labeled “Matched Filtering” has been produced by the PyCBC pipeline and serves as a baseline.

In conclusion, for the parameter space under consideration, the single-detector networks retain performance closely following matched-filter methods even on long duration continuous datasets and when considering FAR thresholds down to once per month. While more sophisticated ML-based methods exist, we believe there is still much to be learned from these first steps.

Chapter 5

First machine learning gravitational-wave search challenge

The contents of this chapter describe a project conducted by a collaboration of the Albert-Einstein-Institut Hannover (represented by Marlin Schäfer, Alexander Nitz and Frank Ohme) and the FSU Jena (represented by myself and prof. Bernd Brügmann), with Elena Cuoco, Chris Messenger and Eliu Huerta as scientific advisors, and contributions of other participating groups. The project is a mock data challenge called the First machine learning gravitational-wave search challenge (MLGWSC-1), meant to create a unified framework for testing ML-based algorithms for detection of BBH merger waveforms in LIGO data. For its full results and all relevant code and data, see [216, 363].

In the second part of this chapter, we focus on our team’s submission to the MLGWSC-1 and its further development after the end of the challenge. Finally, we apply the updated searches to latest LIGO data. The code used to develop the submission is available in the data release [364].

5.1 Test data

The MLGWSC-1 provides the participants with the scripts to generate test data and evaluate their model. The test data generation script accepts a seed for random number generation, which was unknown until the challenge deadline. Therefore, it is possible to evaluate a submission on test data following the challenge distribution, but not to optimize it over the dataset used for the final evaluation.

The test data consists of two streams, one for each LIGO detector (L1 and H1), and further split into shorter segments. Each dataset is stored in a file of the HDF5 format [365] and contains the two groups titled “L1” and “H1” consisting of multiple

segments denoted by their GPS start time in seconds. The segments have a minimum duration of 2 hours and they consist of detector noise sampled at 2048 Hz and potentially BBH waveform injections. We have decided to split the data into smaller independent segments due to the fact that the sensitivity of a real detector varies over time. This renders certain parts of the data unusable for analysis. Therefore, search algorithms need to be able to handle gaps in the data. Splitting the data into independent segments accounts for these gaps and allows us to use different PSDs in different segments of the data and effectively model the PSD drift.

To properly isolate effects which cause specific algorithms to struggle, the test data exists in four distinct difficulties labeled 1 through 4 from the simplest to the most complex. This complexity is reflected in the distributions of both the background noise and the waveform injections.

All waveforms are generated using IMRPhenomXPHM [260], a phenomenological BBH inspiral-merger-ringdown waveform model. It models the precession in case of misaligned component spins as well as higher modes accurately. The parameter distributions vary between the datasets, going from simple spinless short mergers in stationary Gaussian noise in the first to precessing longer events in the fourth.

The extrinsic merger parameters are drawn from the corresponding astrophysical distributions summarized in Tab. 5.1. The exception is the distance, as instead of d_L we draw the *chirp distance* [366]

$$d_c := d_L \left(\frac{\mathcal{M}_{c,0}}{\mathcal{M}_c} \right)^{5/6}, \quad \mathcal{M}_c := \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}, \quad \mathcal{M}_{c,0} := \frac{1.4}{2^{1/5}} M_\odot, \quad (5.1)$$

where \mathcal{M}_c is the *chirp mass*, and $\mathcal{M}_{c,0}$ is merely a fiducial chirp mass used as a basis for calculation. Note that in contrast to [366] we use the luminosity distance instead of the effective distance as a basis. Sampling the chirp distance instead of injecting uniformly in volume reduces the maximum luminosity distance for systems with low \mathcal{M}_c and increases the number of detectable low mass systems. Thus, we can make statistically meaningful statements about the sensitivity for these systems without a large increase in the amount of data analyzed.

The distributions of component masses and spins, as well as the choice of modes generated, depend on the dataset choice and are listed in the sections below. The lower frequency cutoff is set to 20 Hz. The separation between consecutive merger times is drawn from a uniform distribution over [24 s, 30 s]. Finally, we apply a taper to the start of each waveform.

Parameter	Uniform distribution
Coalescence phase	$\Phi_0 \in [0, 2\pi)$
Polarization	$\Psi \in [0, 2\pi)$
Inclination	$\cos \iota \in [-1, 1]$
Declination	$\sin \theta \in [-1, 1]$
Right ascension	$\varphi \in [-\pi, \pi)$
Chirp distance	$d_c^2 \in [130^2 \text{ Mpc}^2, 350^2 \text{ Mpc}^2]$

TABLE 5.1: Summary of the extrinsic test data injection parameters. The merger times are separated by times from a uniform distribution over [24s, 30s]. The intrinsic parameters are given in the text as they differ between datasets.

Once waveforms are injected into the background noise, the resulting data is called the *foreground*. The foreground and background are stored in the format described above and both are used in the evaluation of submissions. A lower frequency cutoff of 15 Hz is applied to the data in both files, further processing is left to the submitted algorithms and factored into the performance evaluation. In addition, a third file containing information on the injections is also saved and required for evaluation.

5.1.1 Dataset 1

The noise used in the first dataset is Gaussian and stationary, and a single fixed PSD is used. The `aLIGOZeroDetHighPower` PSD [355, 359, 360] is chosen for use in all segments of both detectors. Thus, there are no sharp peaks in the noise distribution, and the noise distribution of each segment is known to the participants.

To generate the waveforms, component masses are drawn from a uniform distribution over $[10M_\odot, 50M_\odot]$ and the higher and lower value are the primary and secondary mass, respectively. All spin components are set to zero and no higher modes are simulated, setting the mode array to $(2, 2), (2, -2)$. The signal length is then on the order of 1 s.

This dataset represents a solved problem as it has been excessively studied in the past [162, 163, 215, 352], and is meant as a starting point where researchers can refer to existing literature. The final challenge set was generated with the seed 1068209514 and a start time of 0.

5.1.2 Dataset 2

In the second dataset, the noise is again Gaussian and stationary. The PSD, however, is not fixed and it is more complex, being derived from real data and containing sharp

peaks. We derived 20 PSDs off of the open data from the O3a observing run [367], which are available to the challenge participants. For each detector, the PSD is picked randomly out of these, and remains fixed through all segments in that detector.

Each binary component is allowed to have a spin aligned with the orbital angular momentum and its z -component is uniformly distributed in $[-0.99, 0.99]$. The mass range is extended to $[7M_{\odot}, 50M_{\odot}]$. Through these changes, the maximum length of waveforms has increased to ≈ 20 s. As in the previous dataset, no higher order modes are included.

This dataset represents a considerable increase in difficulty, as it requires the submissions to estimate the PSD in algorithms which include whitening, and due to the increase in signal duration, which is known to have been problematic in many previous ML algorithms [167, 177, 368]. The final challenge set was generated with the seed 2743406703 and a start time of 0.

5.1.3 Dataset 3

In the third dataset, the procedure to generate noise is the same as in the second, with one major difference: the PSD is picked randomly for each segment of each detector. This introduces an effect somewhat similar to the PSD drift of real detector noise.

The distribution of component masses, and therefore the maximum waveform duration ≈ 20 s, remains the same as in dataset 2. However, the spin distribution is altered to allow for misaligned spins. Their magnitudes follow a uniform distribution over $[0, 0.99]$ and the orientations are isotropically distributed. The spins of both binary components are independent of each other. In addition, all higher-order modes available to `IMRPhenomXPHM` are used: $(2, 2)$, $(2, -2)$, $(2, 1)$, $(2, -1)$, $(3, 3)$, $(3, -3)$, $(3, 2)$, $(3, -2)$, $(4, 4)$, $(4, -4)$ [260]. The maximum length of waveforms is ≈ 20 s.

The challenge of this dataset is intended to be the introduction of precession effects and higher modes, which can substantially alter the waveform morphology. Their inclusion in production searches relying on accurate signal models requires increases in template bank size up to millions of templates [83, 84]. The final challenge set was generated with the seed 470182217 and a start time of 0.

5.1.4 Dataset 4

In the fourth dataset, real detector noise from the O3a observing run [357, 367] was used, obtained through GWOSC [362]. The data quality in LIGO detectors is tracked

through a set of flags [284, 362, 369]. The noise was sampled from the parts of the O3a observing run which had the DATA quality flag (i.e. data is available) and none of the flags CBC_CAT1, CBC_CAT2, CBC_HW_INJ, BURST_HW_INJ (i.e. data from auxiliary channels suggests high data quality and no hardware injections are performed) in both LIGO detectors H1 and L1. In addition, a 20 second window was removed centered on any detection listed in GWTC-2 [70], and any remaining segments shorter than 2 hours are discarded. To allow for different noise realizations based on the seed supplied to the data generation algorithm, the L1 data is shifted by a random time from $[0, 240\text{ s}]$, while the H1 data is kept fixed.

The waveforms are generated in a way identical to the previous dataset. This dataset is intended to be indicative of a real-world search in a parameter space which is currently sparsely searched. As many ML searches have proven to generalize well from Gaussian to real detector noise [164, 173, 192, 370], contributions which perform well on dataset 3 are expected to do so on dataset 4 as well. At the same time, handling short glitches may prove difficult for searches focusing on the merger and ringdown. The final challenge set was generated with the seed 2514409456 and a start time of 0.

5.2 Evaluation

A submission is required to take as input a dataset in the format described in Sec. 5.1, and output a set of events in an HDF5 file [365]. The file has to contain three one-dimensional datasets of equal size:

- “time”, containing the GPS times where the algorithm predicts an injection,
- “stat”, consisting of ranking statistic values of the events,
- “var”, containing the estimated timing accuracies of the events.

The evaluation process includes applying the submission to both the foreground and the background file, and the outputs are called foreground events and background events, respectively. Following Eq. (2.50), the background events are used to determine the relationship between the trigger threshold and the FAR, and following Eqs. (2.53), (2.54), the foreground events are used to estimate the relationship between the trigger threshold and the sensitive distance.

Participants may choose to optimize their submission on any of the 4 datasets described in Sec. 5.1. However, each team is only allowed to submit a single algorithm,

which is then evaluated on all 4 datasets. This tests the ability of the submissions to adapt to varying conditions.

Evaluation datasets were generated by the procedure described in Sec. 5.1 in the length of 30 days = 2592000 s and the submissions were tested on the Atlas computing cluster at the Albert-Einstein-Institut in Hannover. Standardized hardware is used to evaluate the contributions, including 8 nVidia RTX 2070 Super GPUs with 8 GB of VRAM each, as well as an assortment of ML packages, including PyTorch and TensorFlow [216, 363]. To provide a fair evaluation procedure, the evaluation of submissions has been performed using datasets generated by the script supplied by the organizers, with a seed previously unknown to the participants as well as the organizers.

Like in Ch. 4, the final evaluation of submissions is performed using the sensitivity curves expressed through the FAR and sensitive distance (see Sec. 2.4.2). An injection is considered found by a search if there is at least one foreground event within $\pm\Delta t$ of the injection, Δt being the timing accuracy assigned to the event by the search algorithm. The injections are not performed uniformly in volume, as we are drawing the chirp distance rather than luminosity distance. Eq. 2.53 is thus modified to

$$V(\mathcal{F}) \approx \frac{V(d_{\max})}{N_I} \sum_{i=1}^{N_{I,\mathcal{F}}} \frac{V\left(d_{c,\max}\left(\frac{\mathcal{M}_{c,i}}{\mathcal{M}_{c,0}}\right)^{5/6}\right)}{V\left(d_{c,\max}\left(\frac{\mathcal{M}_{c,\max}}{\mathcal{M}_{c,0}}\right)^{5/6}\right)} = \frac{V(d_{\max})}{N_I} \sum_{i=1}^{N_{I,\mathcal{F}}} \left(\frac{\mathcal{M}_{c,i}}{\mathcal{M}_{c,\max}}\right)^{5/2}. \quad (5.2)$$

In addition, the wall-clock runtimes of the submissions are recorded as a secondary evaluation metric.

5.3 Results

By the extended deadline on 14 April 2022, 6 contributions have been submitted to the MLGWSC-1. These comprise 4 based on ML methods, and 2 conventional analyses: the matched-filtering based search PyCBC [51–55], and the loosely modeled search cWB [64–68, 371, 372].

The main results can be seen in Fig. 5.1. Each panel contains the sensitivity curves achieved by the submissions on one of the test datasets. We focus on the low FAR region and truncate the plot at $\mathcal{F} = 1000 \text{ month}^{-1}$ for two reasons. First, standalone search algorithms may only report events with low FARs. Second, at high FARs, a large number of detection originate from false associations, i.e. they are random

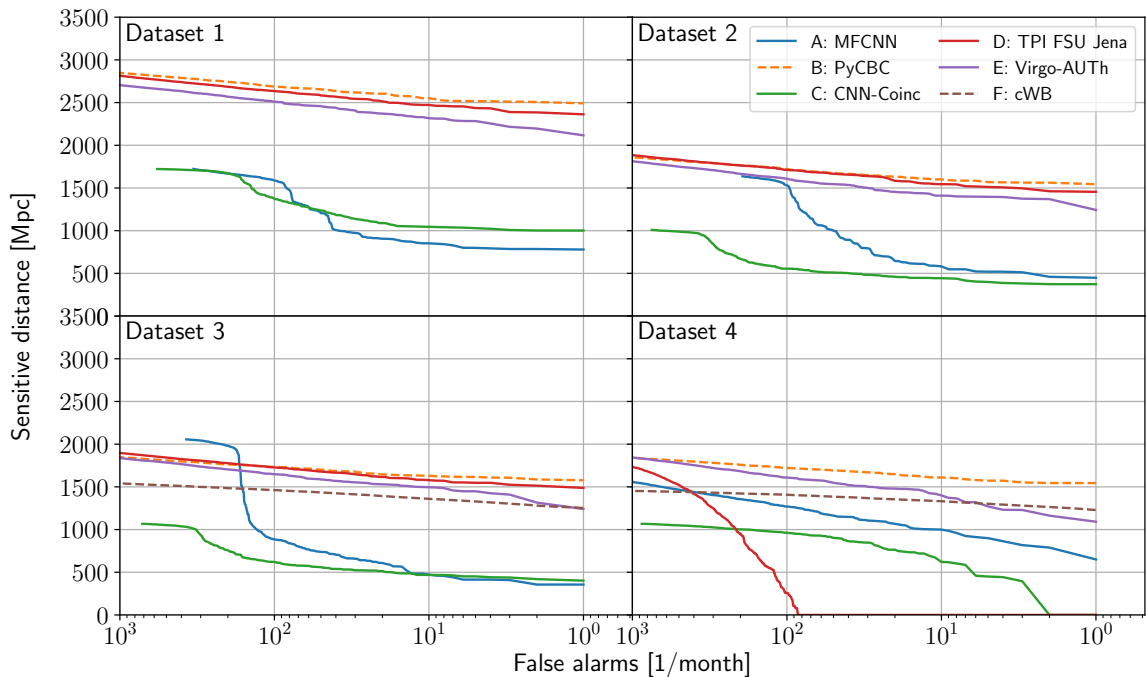


FIGURE 5.1: Sensitivity curves of all submissions over the 4 test datasets. Solid lines refer to ML-based submissions, dashed lines to others. Error bars on the sensitive distance are not plotted as they are smaller than 80 Mpc at every point of every curve and would not be visible.

noise coincidences close enough to an injection and counted as true positives by the evaluation mechanism.

The errors on the sensitive distance estimated from the variance of the sum over randomly drawn waveform parameters are smaller than 80 Mpc over all curves. Due to time constraints, results from the cWB algorithm are only available for datasets 3 and 4. In addition, the PyCBC search used a different template bank for dataset 1 than for the remaining three datasets.

We focus mainly on the two ML-based submissions “D: TPI FSU Jena”, and “E: Virgo-AUTh”. The former is a followup to [163, 215], developed primarily by myself and described in detail in Sec. 5.4, and the latter follows along similar lines. Furthermore, we closely follow the “B: PyCBC” contribution as a matched-filter baseline. For more details on the other submissions and a more detailed discussion of results see [216].

We find that the results of the two submissions are very close to each other in sensitivity on the first three datasets, with the TPI FSU Jena reaching a slightly higher sensitivity at all FARs. At the same time, the Virgo-AUTh submissions retains $\geq 90\%$

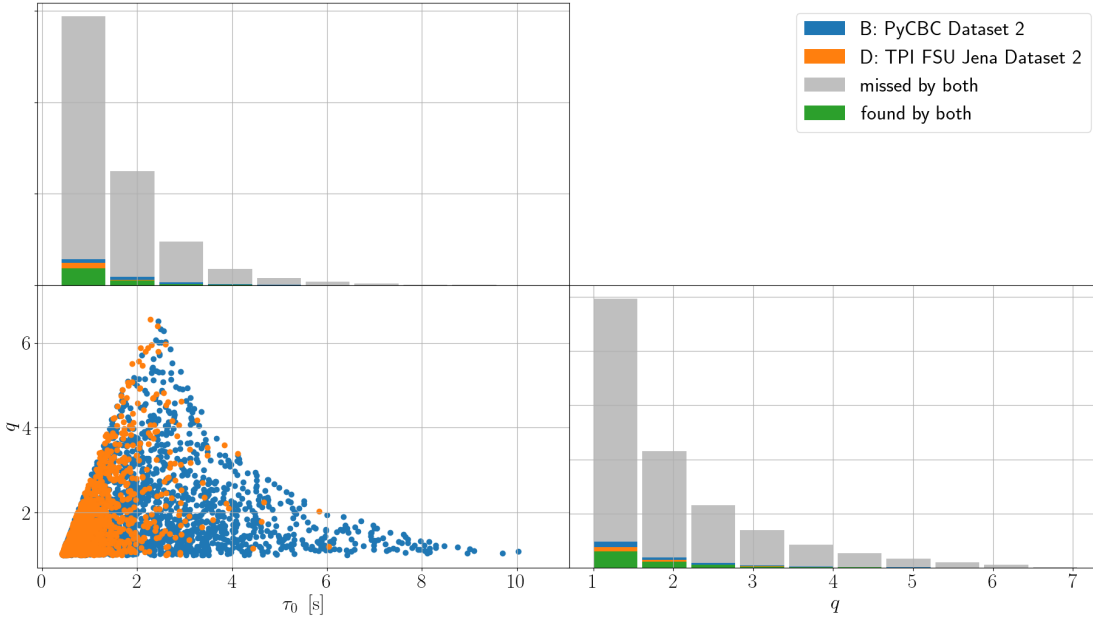


FIGURE 5.2: The injections from dataset 2 identified by the PyCBC and TPI FSU Jena submissions with a $\mathcal{F} \leq 1000 \text{ month}^{-1}$ in the signal duration τ_0 versus mass ratio q plane. The scatter plot shows injections that are found only by one of the two algorithms. Injections that are missed or found by both are only shown in the 1D marginal distribution histograms.

of the sensitive distance achieved by TPI FSU Jena submission for $\mathcal{F} \geq 2 \text{ month}^{-1}$. On dataset 4, however, the Virgo-AUTH submission manages to maintain stable performance, while the TPI FSU Jena contribution is dominated by background triggers and loses all sensitivity at $\mathcal{F} \lesssim 100 \text{ month}^{-1}$. As datasets 3 and 4 have the same distribution of waveforms and the same PSDs and differ merely by the non-Gaussian features of the noise in dataset 4, it is clear that the culprit is a collection of ~ 1000 triggers caused by noise transients.

In comparison to the PyCBC search, on datasets 1-3 the TPI FSU Jena submission is the leading ML search and retains $\geq 93.5\%$ of the sensitive distance achieved by the PyCBC submission at all FARs and at times, even outperforms PyCBC marginally. On dataset 4, the leading ML search is Virgo-AUTH, and it retains $\geq 87\%$ of the sensitivity of PyCBC at $\mathcal{F} \geq 10 \text{ month}^{-1}$. At lower FARs, the sensitivity gap widens and at $\mathcal{F} = 1 \text{ month}^{-1}$, it achieves 70% of the sensitive distance of PyCBC.

Besides studying the sensitivity curves, we also generated found-missed plots for all submissions to highlight specific parameter space areas where the ML searches are competitive and where further development is required. Let us define the mass ratio

$q := m_1/m_2$, and the chirp time parameter

$$\tau_0 := \frac{5}{256 (\pi f_L)^{8/3}} \mathcal{M}_c^{-5/3}, \quad (5.3)$$

where f_L is the lower frequency cutoff of the waveform. The τ_0 is, to first order, the time it takes the binary from orbiting at f_L to merger [373, 374], and allows us to study how well the searches react to waveforms of different lengths.

Fig. 5.2 shows injections in dataset 2 which were identified by exactly one of the PyCBC and TPI FSU Jena submission in the $\tau_0 - q$ space. It shows that the two searches are competitive in the comparable mass region, and the main difference is in the τ_0 distribution of found signals. Most $\tau_0 > 2$ signals found by the TPI FSU Jena search are also found by PyCBC, while the converse is not the case. This indicates that the short duration of the input windows used by the ML algorithm is still a limiting factor. This limitation would likely be more severe if signals from BNS and/or BHNS systems were included.

In this section, we have provided only a superficial summary of the MLGWSC-1 results. For a more detailed analysis of the sensitivities and found-missed plots, see [216]. Detailed plots demonstrating further insights into the parameter space coverage by the different algorithms are available in the data release [363].

As a final note, due to parallelization of the data preprocessing stage and GPU acceleration of the CNN evaluation, the TPI FSU Jena submission is the fastest of all 6, evaluating a full one-month dataset in ~ 1 hour, closely followed by the Virgo-AUTH algorithm at ~ 1.5 hour. The PyCBC search required ~ 43 hours on datasets 2-4 (as it used a different template bank on dataset 1) while using no GPU acceleration and running only on CPUs. At the same time, it is a production search optimized to run on CPUs, making the runtimes inherently incomparable.

5.4 Submission by the TPI FSU Jena team

In this section we focus on the contribution labeled ‘‘D: TPI FSU Jena’’ in more detail. It is based on the results of Ch. 4 and [163, 215] with minor alterations to handle a two-detector input and optimize the network architecture to the problem at hand. All code used to build the submission is available in the data release [364].

Parameter	Uniform distribution
Approximant	IMRPhenomXPHM
Waveform modes	all available (see Sec. 5.1.3)
Component masses	$m_1, m_2 \in [10M_\odot, 50M_\odot]$
Spin magnitudes	$ \chi_1 , \chi_2 \in [0, 0.99]$
Spin directions	isotropic
Coalescence phase	$\Phi_0 \in [0, 2\pi)$
Polarization	$\Psi \in [0, 2\pi)$
Inclination	$\iota \in [0, 2\pi]$
Declination	$\sin \theta \in [-1, 1]$
Right ascension	$\varphi \in [-\pi, \pi)$
Luminosity distance	$d_L = 1 \text{ Mpc}$

TABLE 5.2: Distributions used to draw waveform parameters for the training data.

5.4.1 Training data

In a similar manner to Ch. 4, the training and validation data consist of noise samples, half of which are pure noise and the other half include a waveform injection. They are stored separately, already whitened, to be injected in real time at a varying SNR while training. Since multiple PSDs are used and the PSD is required to whiten the waveform, each waveform is tied to a specific noise sample. In architecture optimization, a smaller dataset of 400 000 Gaussian noise samples, 200 000 of which contain a waveform injection, is used. To train the final submission, we used a dataset of 10^6 real noise samples, $5 \cdot 10^5$ containing a waveform injection. In both cases, the validation dataset is generated in the same way and contains 0.2 as many samples in either class as the training dataset.

To generate Gaussian noise, we use PSDs derived from the real noise file which is part of the MLGWSC-1. The PSD of each dataset is estimated using a segment duration of 1.25 s. For each dataset start time in the original file, we have one PSD per detector. The procedure then repeatedly iterates through the dataset start times and uses the two PSDs to generate the noise in the corresponding detectors. Each noise sample is whitened by the generating PSD using a maximum filter duration of 0.25 s, which leaves us with samples of length 1 s.

Real noise for training the submission is obtained by slicing data from the supplied real noise file. A set of available dataset start times is selected at random and each is whitened by its PSDs, estimated with a segment duration of 0.5 s and maximum filter duration 0.25 s. It is then sliced into 1 s segments with a step size of 0.5 s. These PSDs are also used to whiten the sample's corresponding waveform.

The waveform generation process follows a similar procedure to those in the test datasets 3 and 4. Unlike the single-detector experiment in Ch. 4, we need to sample over the extrinsic parameters which we could previously set to a fixed value. It is possible that the model will learn the effects of the mutual geometrical arrangement of the two detectors, and thus the dataset should comprise injections which corresponds to a single event as registered by the two LIGO detectors.

Tab. 5.2 contains the distributions used for all parameters. Note three differences with respect to the distributions of Tab. 5.1 and Sec. 5.1.3:

- Distance: it is irrelevant as we normalize the waveforms to a fixed network SNR.
- Component masses: instead of $[7M_{\odot}, 50M_{\odot}]$, they are drawn from $[10M_{\odot}, 50M_{\odot}]$ for consistency with the experiments of Ch. 4. In addition, the results are expected to deteriorate when the network with an 1 second input window is presented with signals up to ≈ 20 s. A set of additional training run demonstrates that the effect on the results is marginal and does not lead to a systematic improvement. (see App. A).
- Inclination: instead of the astrophysical distribution $\cos \iota \in [-1, 1]$, we have used $\iota \in [0, 2\pi]$. This is due to an oversight on our part. The dominant effect of the inclination is a multiplicative rescaling of the amplitude and is irrelevant since we are normalizing the waveforms to a fixed network SNR, and the waveform morphology is affected only through higher modes. While these are included in test datasets 3 and 4, the effect is expected to be negligible and a separate training run with the correct astrophysical distribution confirmed that this does not affect the network performance.

Each waveform is generated using `pycbc.waveform.get_td_waveform` and projected onto the two LIGO detectors. The merger time is drawn uniformly from $[0.6\text{ s}, 0.8\text{ s}]$ within the input window and the rest is clipped while leaving an extra 0.125 s on both sides. The network SNR is computed (cf. Eq. (2.48))

$$\rho_{\text{net}} = (h_{\text{H1}}|h_{\text{H1}}) + (h_{\text{L1}}|h_{\text{L1}}) \quad (5.4)$$

and the waveform is whitened using the same PSD as the corresponding noise sample, cropping the excess time on both ends due to wraparound corruption, and divided by the computed ρ_{net} to obtain a waveform with $\rho_{\text{net}} = 1$. Waveforms are injected

into the background noise at runtime at optimal network SNRs drawn uniformly from [7, 20] and the SNRs are generated every epoch.

For testing the submission and its follow-up experiments, datasets 1-4 were generated using the script supplied by the challenge organization team, in a length of 30 days and a start time of 0. For all four, the seed value 04261537 was used. In the following, these are referred to as the *development* datasets.

5.4.2 Architecture

The architecture used is based on that of Ch. 4, slightly restructured and optimized. First, the number of input channels of the first convolutional layer is changed from one to two. As in Ch. 4, the Softmax is used for training and the USR for evaluation; to allow for a more robust usage of this method, however, the Softmax layer has been removed (without applying the USR) and the resulting CNN is called the *base network*. For further details, see the data release [364].

The architecture has been optimized by varying the overall width and depth of the network using a set of 4 procedures described in Tab. 5.3. A more thorough variation of different layer parameters separately has not been performed.

The base network architecture has been optimized by training each of the networks in Tab. 5.4 on a training dataset consisting of 400 000 noise samples and 200 000 waveforms. The noise is Gaussian, using the PSDs of the real noise file included in the MLGWSC-1. The waveforms are generated and processed in the same way as above. The Adam optimizer [314] was used with the learning rate $\eta = 10^{-5}$ and the default values $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a batch size of 32. Each network was trained for 100 epochs, its state was saved at the end of each epoch, and the one with the lowest validation loss was selected for evaluation.

Fig. 5.3 shows the sensitivity curves of all network architectures in Tab. 5.4 evaluated on the development datasets 3 and 4. On both datasets, all networks perform fairly similarly except `half_depth`, which is less sensitive. At a closer look, one can isolate that the `double_full_depth` performs the best on dataset 3, with `double_depth` nearly indistinguishable and `quadruple_depth` a close third, and on dataset 4, `quadruple_depth` performs the best with `double_depth` a close second. Based on this, we have chosen the `double_depth` architecture described in Tab. 5.5 for the final submission.

The network is then trained with a Softmax layer appended using one-hot labels (1, 0), (0, 1) on the training dataset described in Sec. 5.4.1 using real noise and a total

layer	kernel	channels		layer	kernel	channels
Conv1D	K	$M \mapsto N$	\mapsto	Conv1D + ELU	$1 + \lfloor K/2 \rfloor$	$M \mapsto M'$
				Conv1D	$K - \lfloor K/2 \rfloor$	$M' \mapsto N$
Dense		$M \mapsto N$		Dense + Dropout + ELU		$M \mapsto M'$
				Dense		$M' \mapsto N$

(A) Double depth procedure. For the first convolutional layer in the network it holds $M' = N$, otherwise $M' = M$. For the last dense layer in the network $M' = M$, otherwise $M' = N$.

layer	kernel	channels		layer	kernel	channels
Conv1D + ELU	K_1	$M \mapsto M'$	\mapsto	Conv1D	$K_1 + K_2 - 1$	$M \mapsto N$
Conv1D	K_2	$M' \mapsto N$				

(B) Half depth procedure. Dense layers remain unchanged.

layer	kernel	channels		layer	kernel	channels
Conv1D	K	$M \mapsto N$	\mapsto	Conv1D	K	$2M \mapsto 2N$
Dense		$M \mapsto N$			Dense	

(C) Double width procedure. The first and last layer of the network retain their $M = 2$ and $N = 2$, respectively.

layer	kernel	channels		layer	kernel	channels
Conv1D	K	$M \mapsto N$	\mapsto	Conv1D	K	$\lfloor M/2 \rfloor \mapsto \lfloor N/2 \rfloor$
Dense		$M \mapsto N$			Dense	

(D) Half width procedure. The first and last layer of the network retain their $M = 2$ and $N = 2$, respectively.

TABLE 5.3: Summary of hyperparameter variation procedures used to optimize the submission network architecture.

name	Conv1D	Dense
original		
half_width	half width	half width
half_depth	half depth	
double_width	double width	double width
double_depth	double depth	
double_full_depth	double depth	double depth
quadruple_depth	$2 \times$ double depth	

TABLE 5.4: Architectures tested in hyperparameter optimization. The second and third column refer to procedures given in Tab. 5.3 on the convolutional and fully connected layers, respectively. On the last line, $2 \times$ corresponds to two subsequent applications of the procedure.

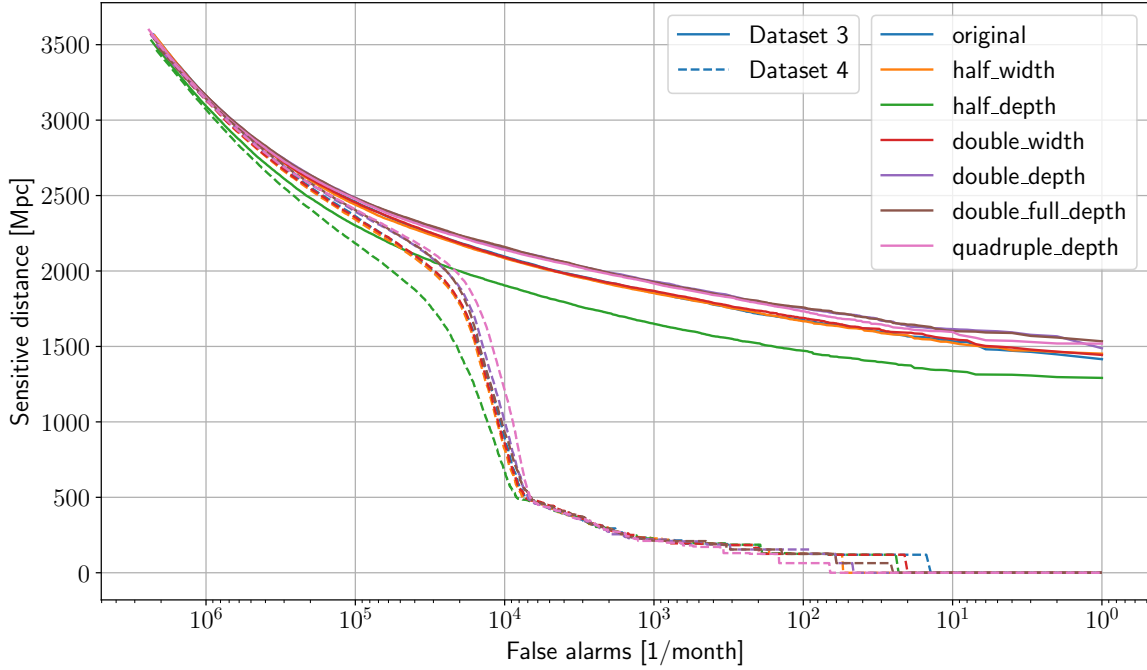


FIGURE 5.3: Sensitivity curves of network architectures of Tab. 5.4 on datasets 3 and 4.

of 10^6 samples, half of which contain an injected waveform. The Adam optimizer [314] was used with the learning rate $\eta = 4 \cdot 10^{-6}$ and the default values $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a batch size of 32, for 250 training epochs. As the loss function, we chose the regularized BCE (see Eq. (4.2)) with $\varepsilon = 10^{-6}$. Gradient $p = 2$ norms were clipped to a maximum of 100 over the entire network at each training batch.

To evaluate the network on continuous data, the sliding window approach is used. As the test data of the MLGWSC-1 consists of segments, each segment is whitened separately using a sample duration of 0.5 s and filter duration 0.25 s, as in the noise sliced into training data. The data is then sliced into 1 second slices at a step size of 0.1 s. The network is applied to the slices using the USR, by taking $x_0 - x_1$, $x_{0,1}$ being the base network outputs. Times where the output exceeds a threshold t are considered first-level triggers and they are assigned a GPS time at 0.6 seconds through the input window. The first-level triggers are clustered into events with a maximum separation of 0.3 seconds, and the events are assigned the GPS time and ranking statistic at the first occurrence of the maximum ranking statistic in the cluster. The timing accuracy of all events is set to a fixed value of 0.2 s to match the size of the merger time window in the training injections.

To choose a suitable value for the first-level trigger threshold t , we decided to optimize for a maximum number of triggers. At very low thresholds, we expect a

layer type	kernel size	output shape
Input + BatchNorm1D		2×2048
Conv1D + ELU	33	16×2016
Conv1D + ELU	32	16×1985
Conv1D + ELU	17	16×1969
Conv1D	16	16×1954
MaxPool1D + ELU	4	16×488
Conv1D + ELU	17	16×472
Conv1D + ELU	16	32×457
Conv1D + ELU	9	32×449
Conv1D	8	32×442
MaxPool1D + ELU	3	32×147
Conv1D + ELU	9	32×139
Conv1D + ELU	8	64×132
Conv1D + ELU	9	64×124
Conv1D	8	64×117
MaxPool1D + ELU	2	64×58
Flatten		3712
Dense + Dropout + ELU		128
Dense + Dropout + ELU		128
Dense		2

TABLE 5.5: Base network architecture chosen for the MLGWSC-1 submission.

sufficient number of first-level triggers to produce a small number of longer events after clustering, and at very high thresholds we expect few first-level triggers and therefore, few events as well. Due to this, we expect the number of triggers as a function of t to have a peak, which could be a suitable value for the submission.

To test this hypothesis and determine the optimal threshold, we applied the trained search at different thresholds to test datasets 3 and 4, generated in a reduced length of 1 day using the seed 14021995. The resulting sensitivity curves as well as numbers of triggers are shown in Fig. 5.4. The figure confirms a peak at $t = -8$ on both datasets on foreground as well as background data, which was subsequently chosen for the submission to the MLGWSC-1. However, the sensitivity curves suggest that at low FARs, the algorithms' performance is insensitive to the choice of the threshold.

For reproducibility, we have used a test dataset generated by the public resources of the MLGWSC-1 whose noise overlaps with that of the training data to present our results. To ensure the consistency of the experiments, a second dataset has been used following the same distribution but excluding the segments used in the previously applied test data as well as the training data (see Sec. 5.6).

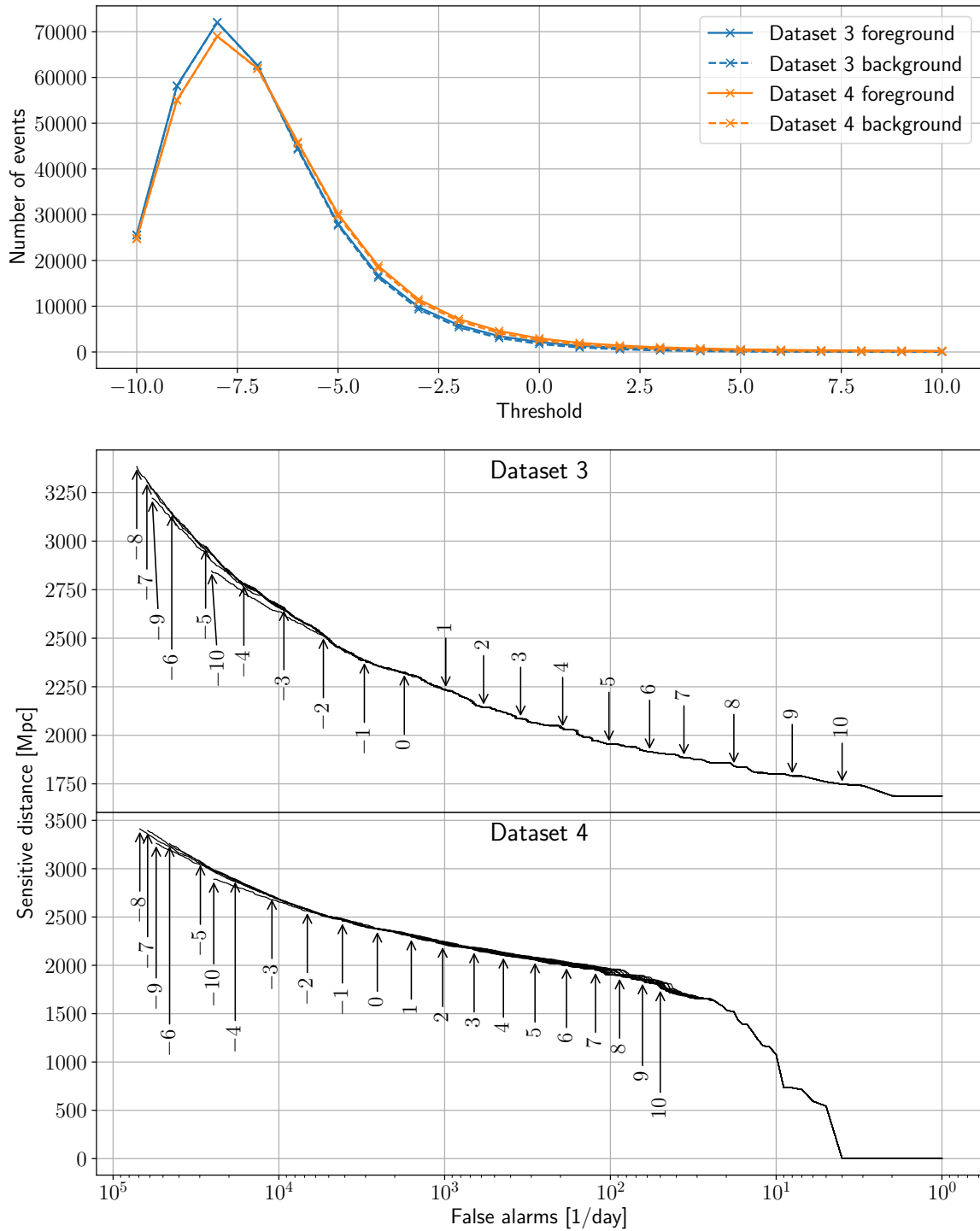


FIGURE 5.4: Numbers of triggers (top panel) and corresponding sensitivity curves (bottom panels) of the search when integer first-level trigger thresholds $t \in [-10, 10]$ are used. Testing is performed on test datasets 3 and 4 of reduced length 1 day. Due to large overlap between the sensitivity curves, rather than color-coding, their left ends are annotated with the threshold value. All curves reach the same point at the right end $\mathcal{F} = 1 \text{ day}^{-1}$.

5.5 Correction of performance issues

As seen in Sec. 5.3 and shown in Fig. 5.1, the TPI FSU Jena submission, described in this section, has performed comparably to the Virgo-AUTH submission on test datasets 1-3, and lost all sensitivity at $\mathcal{F} \lesssim 10^2$ per month, unlike Virgo-AUTH. The cause of the degradation was a large number of background triggers which raised the threshold at $\mathcal{F} \lesssim 100 \text{ month}^{-1}$ above the ranking statistic of the loudest injection.

The search contains no mechanisms designed to reject triggers from noise transients (besides the exclusion of segments with the `CBC_CAT1`, `CBC_CAT2`, `CBC_HW_INJ`, `BURST_HW_INJ` flags). Instead, it was meant to learn the distinction between noise transients and BBH signals from the learning data. In this section, we make modifications which lead to an improvement in sensitivity by learning to reject short noise transients.

We have isolated 3 differences between our algorithm and that of Virgo-AUTH which may be the cause of the performance gap:

- architecture: The Virgo-AUTH team has used a ResNet-like architecture [123, 375], while we chose the CNN architecture optimized in Sec. 5.4.2.
- data normalization: The Virgo-AUTH team has determined that the deep adaptive input normalization (DAIN) [376] layer leads to better performance than the batch normalization used by us [375, 377].
- dataset composition: The training data used by the Virgo-AUTH team has used non-overlapping slices to obtain noise data for training, while ours overlap halfway, potentially reducing our effective dataset size.

We have attempted to resolve the issue by modifying these aspects of our search individually as well as combined in order to isolate their effects. In experiments using the ResNet-like architecture, we have observed systematically lower performance and longer training times than when using our optimized CNN. Therefore we have decided to keep the architecture used for our original submission.

To test the effect of the normalization, we have implemented the DAIN [376] in our search and evaluated its performance, as well as one where the normalization was removed entirely. Both options improved the search performance; however, higher sensitivities were obtained in the experiments without an input normalization layer.

Finally, we have attempted to increase the size of our dataset as well as increase the percentage of pure noise samples in order to address the issue of the reduced

effective noise dataset size. Surprisingly, experiments with a larger dataset brought no discernible improvement. Instead, the critical step was to build the training data out of 75% pure noise and 25% with injections. Shifting this ratio further led to deterioration again.

Therefore, we have decided to make those two changes to the submission algorithm to improve its performance: remove the normalization layer and change the ratio of classes in the training dataset. We perform the second by using the same generated training dataset and reading in all of the noise samples but only half of the waveforms, i.e. the final dataset consists of 750 000 pure noise samples and 250 000 containing an injection.

Epoch selection

The final issue to tackle at this point is the selection of the optimal network state. While the network state for the submission has been chosen primarily based on low validation loss values, we suspected these may not translate directly to high sensitivity. Since the most relevant FARs are the lowest, we choose the sensitive distance at $\mathcal{F} = 1 \text{ month}^{-1}$ as the main metric to optimize.

The network was trained in 6 independent training runs using the setup at which we arrived above and its sensitivity curves were evaluated at each of the 250 epochs of the training procedure. Fig. 5.5 compares the sensitive distance at $\mathcal{F} = 1 \text{ month}^{-1}$ to the corresponding validation loss value. On dataset 4, it clearly demonstrates that there is no direct relationship between these two metrics and thus choosing the ideal network state based on the loss curve is unreliable.

We therefore choose suitable states based on the sensitive distance on dataset 4 at $\mathcal{F} = 1 \text{ month}^{-1}$ during training. The corresponding sensitivity evolution is displayed in Fig. 5.6. On dataset 3, the sensitivity rises to $\sim 1500 \text{ Mpc}$ in the first ~ 50 epochs in all 6 runs, and remains stable with a minor decline towards the end. On dataset 4, however, there is a large number of jumps which are undesirable, as they suggest a degree of instability in the training procedure.

Unfortunately, we were unsuccessful in resolving this issue. At the same time, each of the 6 runs has reached at least 1149 Mpc in sensitive distance on dataset 4 at $\mathcal{F} = 1 \text{ month}^{-1}$ at some point during the training, and Run 1 even above 1400 Mpc . For reference, the Virgo-AUTH submission reaches a sensitive distance of $\doteq 1092 \text{ Mpc}$ at $\mathcal{F} = 1 \text{ month}^{-1}$ on the challenge dataset 4. We therefore conclude that despite the

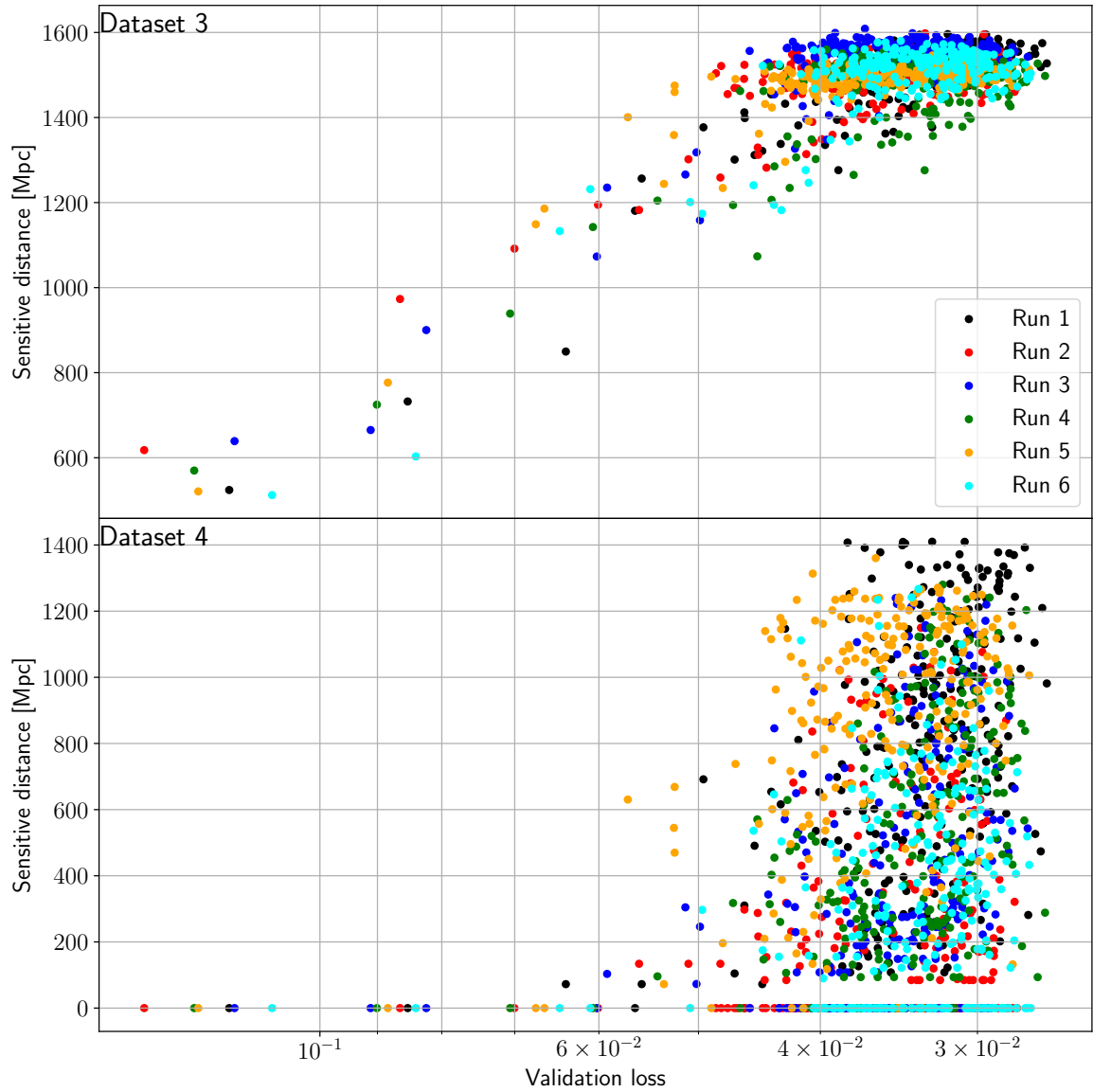


FIGURE 5.5: Scatter plot of the validation loss vs. the sensitive distance at all training epochs for a set of 6 training runs. The sensitive distance in the top and bottom panels is evaluated on datasets 3 and 4, respectively, at $\mathcal{F} = 1 \text{ month}^{-1}$.

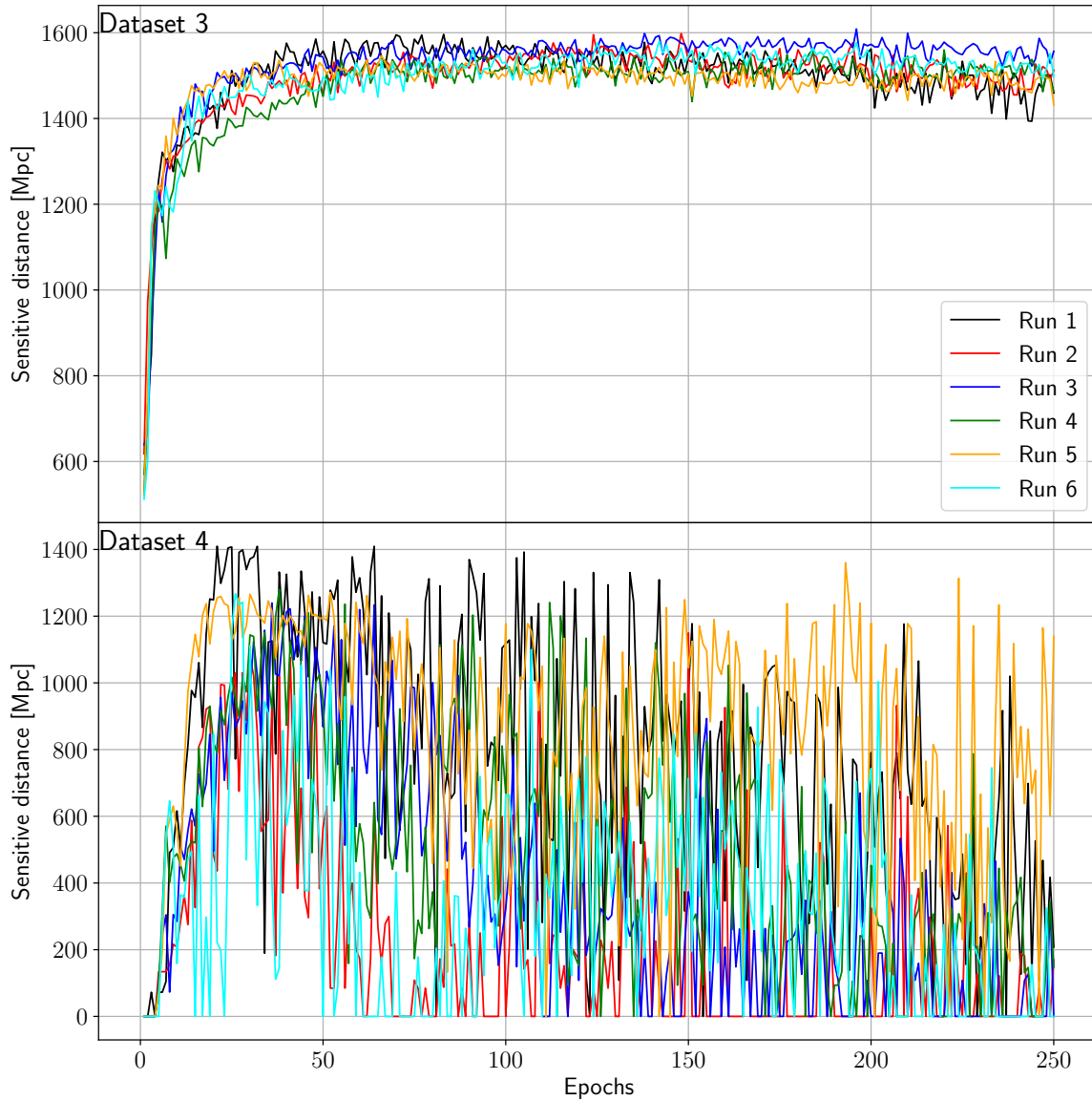


FIGURE 5.6: Evolution of the sensitive distance during training for a set of 6 training runs. The sensitive distance in the top and bottom panels is evaluated on datasets 3 and 4, respectively, at $\mathcal{F} = 1 \text{ month}^{-1}$.

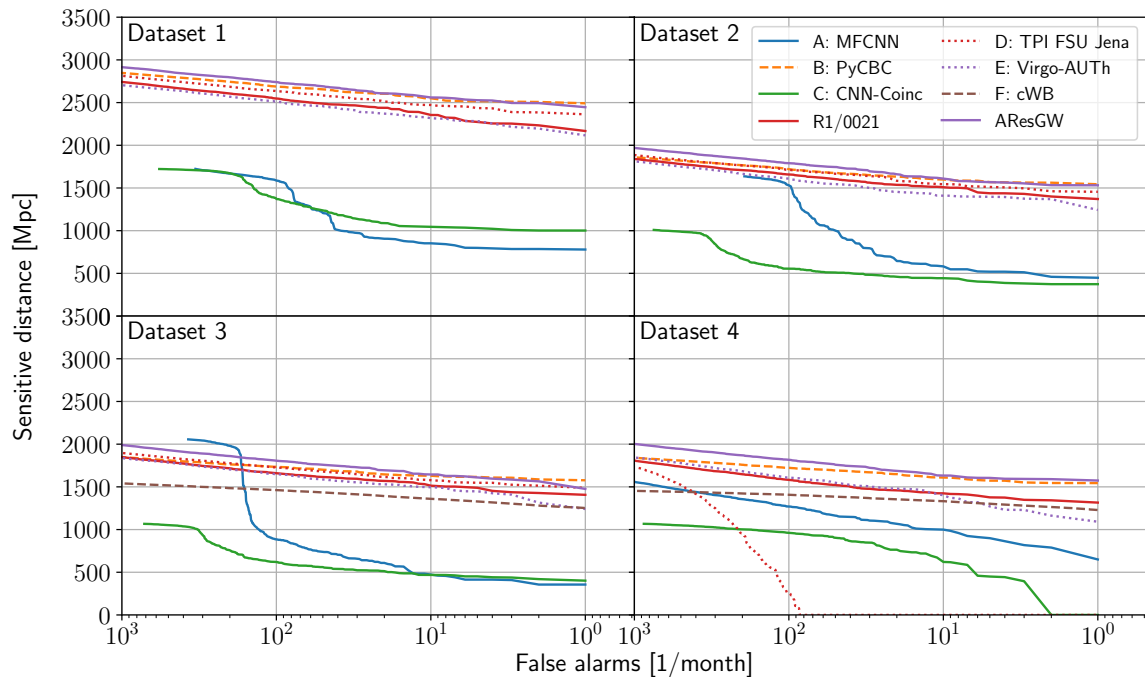


FIGURE 5.7: Sensitivity curves of all original submissions over the 4 test datasets used to evaluate the MLGWSC-1. The R1/0021 algorithm replaces our original submission, and the AResGW replaces the Virgo-AUTH submission. The original contributions are represented by dotted lines instead.

sensitivity oscillations, our training procedure reliably leads to well-trained networks when a detailed state selection procedure is employed.

In the following, we select the states at the global maxima of the curves of Fig. 5.6 and apply them to more data. For this purpose they are referred to in the format R<run number>/<four-digit epoch number>. As the maximum over all 6 curves is reached by the first run at epoch 21, the corresponding state R1/0021 is the one we emphasize the most. As such, we evaluate it on the challenge datasets used to produce Fig. 5.1, and plot the results in Fig. 5.7.

On datasets 1-3 we observe a small decline in sensitivity. This is to be expected, as we modified a setup which was highly efficient on Gaussian noise to optimize it on a different background noise distribution, and the model may now reject injections which are similar to a glitch. Despite this decline, our search retains its position as the leading ML submission.

On dataset 4, our updated search performs marginally worse than the Virgo-AUTH submission at $\mathcal{F} \gtrsim 10 \text{ month}^{-1}$. At lower values, the Virgo-AUTH search declines faster than previously, while R1/0021 remains more stable and reaches a FAR of 1 per month

at a sensitive distance ~ 1300 Mpc, closely beating the cWB algorithm as well, while still underperforming in comparison to the PyCBC search.

It should be emphasized at this point that our updated search had much more time for development as well as information about both the Virgo-AUTH search and the overall results of the challenge. In the same spirit, the Virgo-AUTH team has also improved the performance of their search under the name AResGW, narrowly beating the PyCBC submission on dataset 4 at all FARs. On datasets 1-3, AResGW also outperforms the PyCBC search at $\mathcal{F} \gtrsim 5 \text{ month}^{-1}$ while performing marginally worse at FARs down to 1 per month. For further details, we refer the reader to [375, 377].

5.6 Validation

Following Sec. 3.2, a ML algorithm must be independently evaluated on previously unused test data. In the framework of the MLGWSC-1, when test data of dataset 4 is generated, changing the seed value alters the noise by a time shift. One may therefore make the case that the noise data generated at different seeds is mutually correlated, which introduces a bias in the evaluation procedure.

During the development of the original submission, a test dataset was used for optimization of the algorithm. After doing so, it is necessary to validate it on a fully independent dataset to prevent overfitting. We ensure this by only using those segments from `real_noise_file.hdf` which neither appear in the test dataset used for development nor have been sliced into training data. We generate the data with a seed value of 1935132042 and a start time of 0 in a length of 3 days and call the resulting dataset *cropped*, for details see App. B.

To ensure that the submission did not suffer from overfitting, we apply it to the cropped dataset and compare the sensitivity curve to the one obtained on the dataset used in development in the top panel of Fig. 5.8. There is a minor decline in sensitivity at $\mathcal{F} > 100 \text{ month}^{-1}$, which is expected, and the gap closes when approaching this value. At $\mathcal{F} \lesssim 100 \text{ month}^{-1}$, the sensitivity is zero in the same way as the on the development dataset.

In addition, to ensure the robustness of our improved algorithm as described in Sec. 5.5, we also evaluate the networks obtained at peak sensitivity of Fig. 5.6 on the cropped dataset. The resulting sensitivity curves can be seen in the bottom panel of Fig. 5.8. These also suggest that a potential overfitting issue is negligible, as the curves obtained on the development and cropped datasets match each other quite

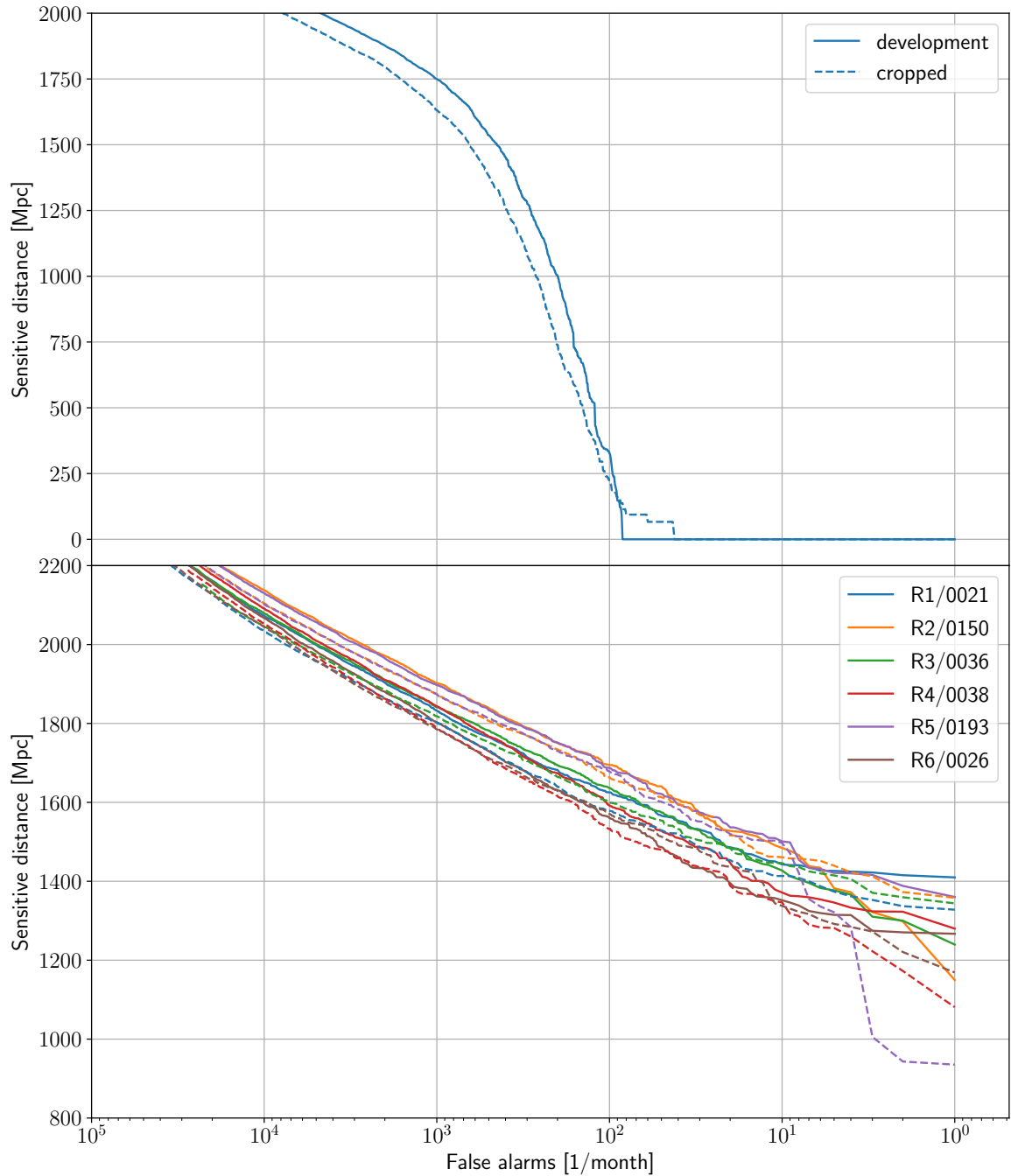


FIGURE 5.8: Sensitivity curves of the submission (top panel) and the 6 network states obtained as maxima of the curves in the lower panel of Fig. 5.6 (bottom panel). The solid lines have been evaluated on the dataset used for development, while the dashed lines have been evaluated on a month-long dataset 4, generated excluding the noise segments which appear in the challenge dataset 4, as well as in the training data. The top panel's legend transfers to the bottom panel as well, i.e. a solid and a dashed line of the same color are the development and cropped dataset sensitivity curves, respectively, of the network given by the corresponding color in the bottom panel's legend.

closely. The exception is the R5/0193 network at $\mathcal{F} \leq 10 \text{ month}^{-1}$, which experiences a significantly larger decline in this area than others.

Based on the bottom panel of Fig. 5.8, we expect the networks R1/0021, R2/0150, and R3/0036 to perform well, as their sensitive distance on the cropped dataset at $\mathcal{F} = 1 \text{ month}^{-1}$ lies between 1300 Mpc and 1400 Mpc. At the same time, it should be noted that besides this inference, results obtained on the cropped dataset had no influence on any other results in this thesis.

5.7 Application to O3b data

The O3 LIGO observing run was split by a commissioning break into two phases, O3a and O3b [357, 362]. The first part was used through this chapter to train as well as test CNN-based searches for BBH merger waveforms. In this section, we apply the searches developed in Sec. 5.5 to real data recorded by LIGO through the O3b phase and cross-reference the output with the transients recorded in the GWTC-3 catalog [72].

The script used to query O3a data and subsequently process it into the noise file used to generate the noise of dataset 4 of the MLGWSC-1 has been modified to fetch O3b data and run including known events and a minimum segment duration of one minute. The resulting file consists of 2377 segments, totaling 8228706 seconds of data. This corresponds to roughly 95 days and 6 hours out of the full 147 days and 2 hours forming the O3b observing run, as we only use segments where data from both LIGO observatories is available in sufficient quality (see Sec. 5.1.4).

All 6 searches corresponding to the highest peaks of each line in the bottom panel of Fig. 5.6 have been applied to this data. The GWTC-3 catalog [72] consists of 35 confident detections and 7 marginal. Events lying outside the segments of available data are excluded, leaving us with 31 confident and 4 marginal events to be found. In addition, an event which occurs close to the beginning or the end of the segment may be more difficult to identify, and we have verified that this does not occur. In fact, all events were at least two hundred seconds from either end of their corresponding segment, with the exception of GW200316_215756, which occurred 46.1 seconds after its segment began.

Although the maximum NS mass is currently uncertain, estimated to be between $2.1M_{\odot}$ and $2.7M_{\odot}$ [378–381], following [72] we use $3M_{\odot}$ as a robust upper limit. Of the 35 confident detections, 32 are unambiguous BBHs, as they are estimated at

$m_2 > 3M_\odot$ at 97% probability. The three remaining events are GW191219_163120, GW200115_042309, and GW200210_092254, and they are BHNS merger candidates. All three are among the 31 events in the analyzed segments. None of the GWTC-3 events are BNS candidates [72].

Tab. 5.6 shows FARs at which the searches under evaluation detected events from the GWTC-3-confident catalog. An event is considered detected by a search if it triggered within 0.2 seconds of the merger time given in the GWTC-3 catalog at a FAR lower than 1000 per month. The FARs are estimated as the number of false alarms assigned a higher ranking statistic than the corresponding event, divided by the total length of analyzed data. It should be noted that at very low FAR values down to zero, these values are inaccurate as the noise distribution has not been sampled well in this region, and are to be taken with a grain of salt. None of the 4 GWTC-3-marginal events in the available segments were found by either of the searches at $\mathcal{F} < 1000 \text{ month}^{-1}$.

First, let us focus on the 9 events in the top section, i.e. those whose 90% credible intervals on component masses lie completely in the $[10M_\odot, 50M_\odot]$ range used to generate training data. 6 of them are identified at $\mathcal{F} \leq 5 \text{ month}^{-1}$, in most cases $\sim 1 \text{ month}^{-1}$, by all 6 evaluated searches. Let us comment shortly on the remaining 3:

- **GW200208_130117** This event is found by all searches at $\mathcal{F} < 20 \text{ month}^{-1}$, and by 4 at $\mathcal{F} < 4 \text{ month}^{-1}$.
- **GW200219_094415** This event is found by all searches at $\mathcal{F} < 40 \text{ month}^{-1}$, and by 3 at $\mathcal{F} < 10 \text{ month}^{-1}$.
- **GW200308_173609** This event is missed by all searches. At $\rho_{\text{net}} \doteq 7.1$, it is the second faintest event in GWTC-3. It was only marked confident by the pycbc_bbh search at $p_{\text{astro}} = 0.86$, while other searches reported $p_{\text{astro}} < 0.5$. According to the catalog's parameter inference pipeline, this is due to a high distance from Earth at $d_L = 5.4_{-2.6}^{+2.7} \text{ Gpc}$ (median and 90% credible intervals), the largest median luminosity distance in the catalog.

Event name	ρ_{MF}	R1/0021	R2/0150	R3/0036	R4/0038	R5/0193	R6/0026
		\mathcal{F} [month ⁻¹]					
GW200224_222234	20.0	0.0	0.9	0.0	0.0	0.0	0.0
GW200311_115853	17.8	0.0	0.9	0.6	1.3	0.0	6.0
GW200225_060421	12.5	0.0	1.6	0.0	0.0	1.9	0.3
GW191215_223052	11.2	0.0	1.9	1.3	1.3	0.0	1.3
GW200208_130117	10.8	19.2	2.2	3.5	3.1	1.6	10.1
GW200219_094415	10.7	5.0	4.7	8.8	38.7	12.6	19.5
GW200209_085452	9.6	1.3	2.8	0.9	2.8	2.2	0.3
GW191204_110529	8.8	1.6	3.1	0.0	3.1	5.0	3.1
GW200308_173609	7.1	-	-	-	-	-	-
GW191222_033537	12.5	0.0	4.1	2.5	2.5	0.3	0.3
GW200128_022011	10.6	25.5	3.1	0.0	11.7	10.4	2.2
GW191230_180458	10.4	6.6	149	19.5	98.9	36.9	5.0
GW191127_050227	9.2	38.7	2.8	4.4	18.6	3.1	6.6
GW200220_124850	8.5	215	517	956	96.1	695	375
GW191126_115259	8.3	-	-	-	-	-	-
GW200216_220804	8.1	-	189	-	-	841	-
GW191113_071753	7.9	-	634	391	-	713	647
GW200306_093714	7.8	485	407	720	-	69.3	-
GW200208_222617	7.4	38.1	6.0	19.5	55.1	159	187
GW200322_091133	6.0	810	898	-	-	-	-
GW191204_171526	17.5	3.5	8.8	4.1	4.4	7.6	6.0
GW191109_010717	17.3	0.0	1.9	0.9	0.6	1.3	0.9
GW191129_134029	13.1	-	-	-	-	-	-
GW200115_042309	11.3	-	-	-	-	-	-
GW200202_154313	10.8	-	-	-	-	-	-
GW200316_215756	10.3	-	-	-	-	-	-
GW191105_143521	9.7	-	-	-	-	658	-
GW191219_163120	9.1	-	-	-	-	-	-
GW191103_012549	8.9	-	-	-	-	-	-
GW200210_092254	8.4	-	-	-	-	-	-
GW200220_061928	7.2	-	-	-	-	-	-

TABLE 5.6: List of O3b events and their identification by the 6 searches of Fig. 5.8. Only events from the `GWTC-3-confident` catalog [72] are included. Events which were not recovered by the given search are marked by a hyphen. The accompanying SNR values come from the catalog’s parameter inference pipeline based on Bilby [78, 79, 382]. In addition, 4 events were omitted due to insufficient data quality or availability.

The events are split into three sections by horizontal lines. The top section contains events whose 90% credible intervals for both component masses lie fully in the interval $[10M_{\odot}, 50M_{\odot}]$ used to generate the training data. For events in the second section, at least one does not fulfill this but both intervals lie at least partly in the training range, and in the third section, at least one of the intervals lies fully outside the training range. Within each section, the events are sorted by their SNRs in a descending order. The credible intervals come from the parameter estimation supplied by GWOSC [383].

In a further 11 events, the 90% credible intervals of both component masses lie at least partly in the training range, and at least one lies partly outside. These comprise the second section of Tab. 5.6. 5 of them are found at $\mathcal{F} < 10 \text{ month}^{-1}$ by at least one search, while the remaining 6 are recovered only at $\mathcal{F} > 96 \text{ month}^{-1}$ or missed completely. Two of the former are found to a comparable degree of confidence as some of the events above:

- **GW191222_033537** This event is found by all searches at $\mathcal{F} \leq 4.1 \text{ month}^{-1}$. Its component masses are estimated at $45.1_{-8.0}^{+10.9} M_{\odot}$, $34.7_{-10.5}^{+9.3} M_{\odot}$, putting the secondary mass interval fully into the training range, and just under half of the primary as well. At $\rho_{\text{net}} \doteq 12.5$, it is one of the louder events in the available segments and the loudest of events in the second section of Tab. 5.6.
- **GW191127_050227** This event is found by all searches at $\mathcal{F} < 40 \text{ month}^{-1}$, and by 3 at $\mathcal{F} < 5 \text{ month}^{-1}$.

In the remaining 11 events, in the third section, the 90% credible interval of at least one component mass lies fully outside the training range. 8 are completely missed, including all three BHNS candidates. One event is found by a single search at $\mathcal{F} \doteq 658 \text{ month}^{-1}$, and the remaining two are recovered by all searches:

- **GW191204_171526** This event is found by all searches at $\mathcal{F} < 10 \text{ month}^{-1}$. The secondary mass, estimated at $m_2 = 8.2_{-1.6}^{+1.4} M_{\odot}$, lies just under the lower end of the training range.
- **GW191109_010717** This event is found by all searches at $\mathcal{F} < 2 \text{ month}^{-1}$. Its primary mass lies above the upper end of the training range, estimated at $m_1 = 65_{-11}^{+11} M_{\odot}$.

They are the third and fourth loudest, respectively, events in the analyzed segments, and the loudest in the third section of Tab. 5.6. This demonstrates that our networks are sensitive to louder events even outside the parameter space region used for training.

To summarize, the searches can reliably identify $\rho_{\text{net}} \geq 8$ events in the parameter space area used to train them at low FARs, as well as the louder events outside. To illustrate the search results, we present Q-transform spectrograms of some of the loudest events returned by the R1/0021 search. Fig. 5.9 shows the two loudest events, which we identified as the BBH mergers GW200225_060421 and GW200224_222234. Fig. 5.10 shows the two loudest false alarms.

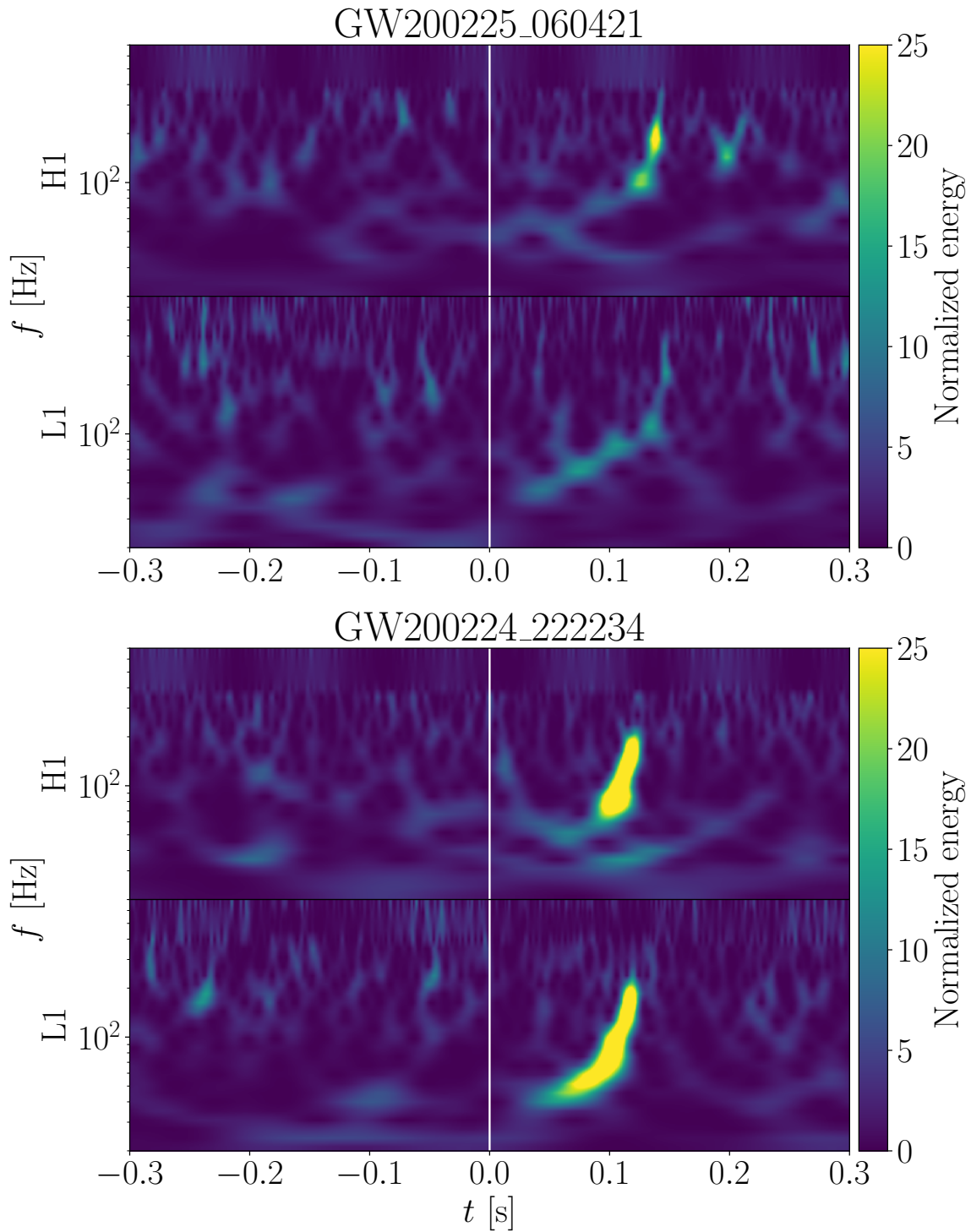


FIGURE 5.9: Q-transform plots of the two loudest O3b events as returned by the R1/0021 search. The full vertical white line denotes the time returned by the search and the time axis is shifted to zero at this time.

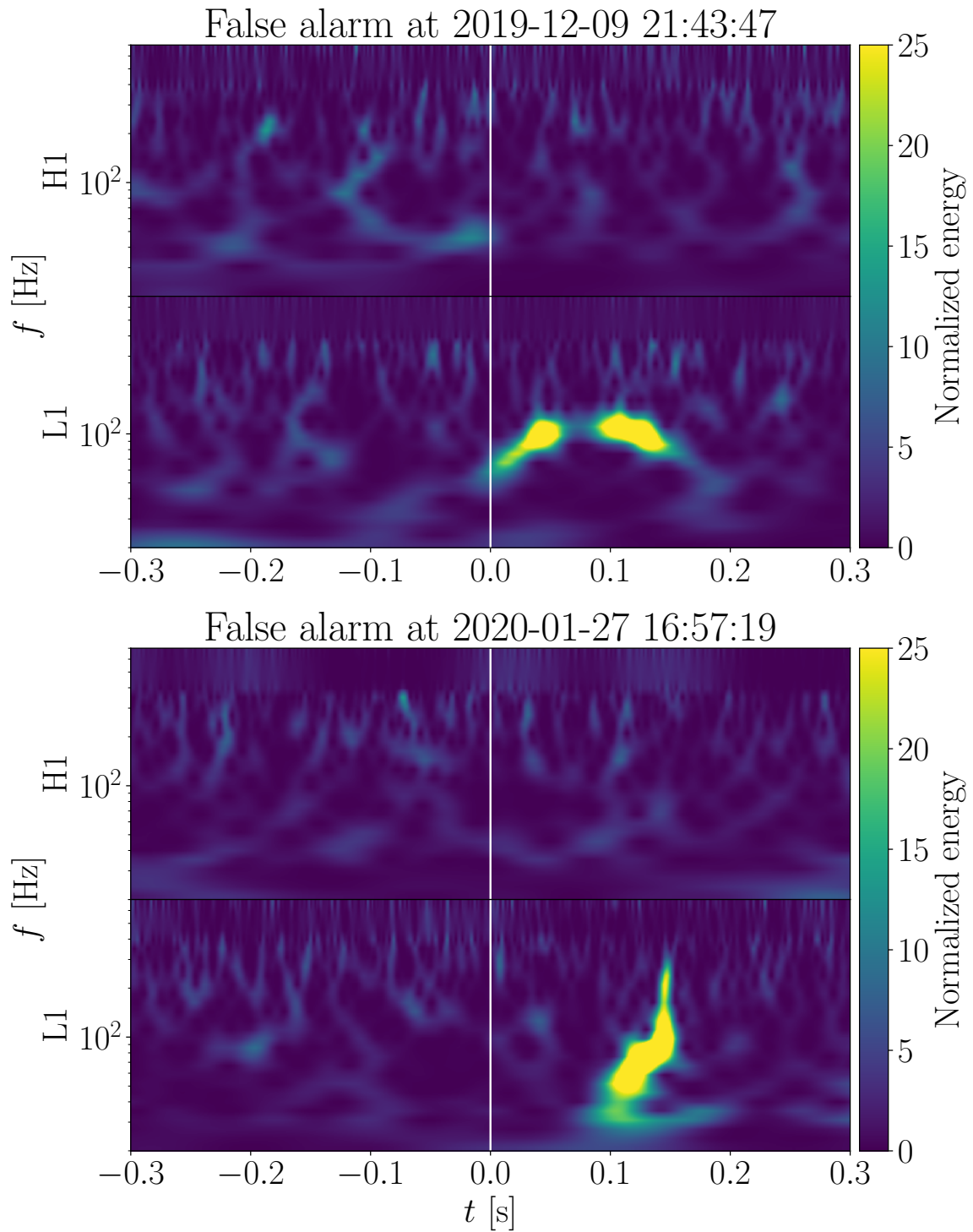


FIGURE 5.10: Q-transform plots of the two loudest O3b false alarms as returned by the R1/0021 search. The full vertical white line denotes the time returned by the search and the time axis is shifted to zero at this time.

Spectrograms of the loudest 128 events returned by each of the 6 searches analyzed in Tab. 5.6 annotated by the corresponding event name or time in case of a false alarm are available in [364]. They clearly demonstrate that a majority of loud false alarms feature a glitch in their spectrogram, often one which resembles the characteristic chirp observed in CBCs. More specifically, looking at the highest significance events from each run, it seems by eye that different searches are sensitive to different glitch types (see Fig. 2.5):

- R3/0036 and R4/0038 mainly to the “whistle” glitches,
- R2/0150 and R5/0193 mainly to the “extremely loud” glitches,
- R1/0021 and R6/0026 to a mixture of the ones above, as well as “chirp” or “blip” types.

Chapter 6

Conclusion

In this dissertation, several results on using ML-based algorithms to detect GW signals radiated by BBHs were developed and presented. Namely, we have explored efficient ways of training CNN-based searches on LIGO data, examined their crucial properties, and developed methods of correcting some issues which may arise. Especially, the methods introduced here pushed FARs down to production levels of 1 per month, demonstrating that they are ready for real world deployment in the tested scenarios.

In the project described in Ch. 4 [215], we have followed up the pioneering applications of ML methods to GW data [162, 163]. First, we reproduced their results in classifying isolated GW detector data samples, and extended them to a lower FAR, which is critical for GW searches. In the process of doing so, we developed the USR, a method to resolve a particular numerical issue which occurred frequently in our experiments and prevented evaluation at low FARs.

Furthermore, we tested multiple methods of curriculum learning, a method of increasing the training dataset’s complexity as the training process progresses. We have concluded that the crucial part is the inclusion of fairly weak signals in the training data, and there was no benefit in introducing them later during the training process. We observed that networks which were presented with weak signals during training could generalize to stronger signals during testing, while the opposite is not the case.

Following this, we have applied the networks trained in the first part of this project to search for BBH signals in continuous data. By applying the sliding window approach, we have extended these networks into single detector search algorithms which remain sensitive down to a FAR of 1 per month. A comparison with an equivalent matched-filtering pipeline, namely the PyCBC search [51–55], shows that our searches retain $\geq 91.5\%$ of the conventional pipeline down to $\text{FAR} = 1 \text{ month}^{-1}$. We have performed all experiments of this project in both PyTorch [124] and TensorFlow [125, 126] with virtually indistinguishable results.

In the project described in Ch. 5, we followed up the previous project by organizing a mock data challenge titled MLGWSC-1, focusing on searches for BBH waveforms in continuous segments in two detectors [216]. It included multiple test datasets following different distributions for both the waveform injections and the background noise, which allowed us to better understand the impact of effects such as precession, a longer inspiral time, or differing levels of noise realism on the submissions' performance. The noise in the final dataset consisted of real detector noise from the O3a observing run, including noise transients.

The MLGWSC-1 attracted 4 ML-based submissions and 2 conventional search algorithms as baselines (a targeted BBH search, and a loosely modeled search). One of the ML searches was developed by our group and is a coherent search based on Ch. 4 and trained using real O3a noise. On the datasets built upon Gaussian noise, our submission has reached the highest sensitivities, operating at $>90\%$ of the sensitive volume of the PyCBC search. In addition, due to an efficient parallelization of segment whitening and serialized GPU evaluation of the CNN at the core of our search, we have achieved the lowest runtimes of all MLGWSC-1 submissions on the given hardware.

Unlike conventional searches, we supplied no additional information nor veto systems to reject likely false alarms from noise transients, and let the network learn their morphology by itself. While this was unsuccessful in the original submission and the search results were dominated by triggers from those noise transients, further effort at optimization has allowed us to resolve this issue. By removing the batch normalization layer and optimizing the distribution of the training data, we have managed to surpass all other challenge contributions besides the PyCBC search on real noise, at the cost of a slight loss of sensitivity on the datasets based on Gaussian background noise. At the same time, the leading ML submission has also been improved and surpassed both our updated algorithm and the PyCBC search [375, 377].

Finally, we have applied several of the trained searches to data from the second half of LIGO's third observing run. Out of the known BBH events [72] at which our searches are directed (i.e. those with both component masses in the $[10M_{\odot}, 50M_{\odot}]$ range) and which were in the analyzed segments, our searches detected all but one at $\text{FAR} < 10 \text{ month}^{-1}$, the exception being a very weak signal at $\rho_{\text{net}} = 7.1$ due to its large distance from Earth. In addition, we have observed that the trained networks tend to produce the loudest false alarms on different types of background noise transients, suggesting a potential improvement in the future by using an ensemble of networks [173].

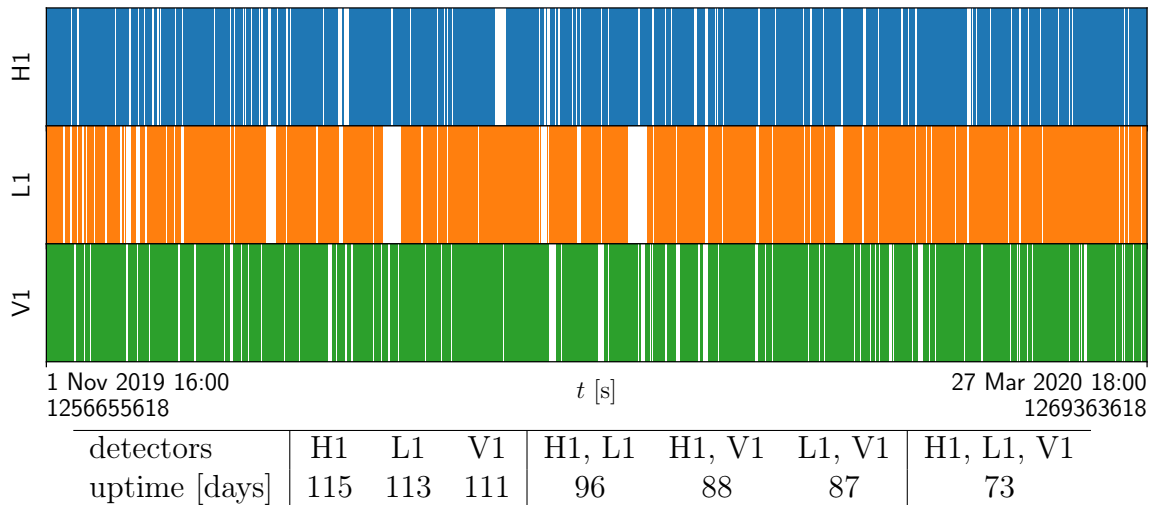


FIGURE 6.1: Overview of segments with available data of sufficient quality in the H1, L1 and V1 detectors. Following Sec. 5.1.4, this refers to segments with the `DATA` flag and none of the `CBC_CAT1`, `CBC_CAT2`, `CBC_HW_INJ`, `BURST_HW_INJ` flags in the detector in question. Top: bar plot demonstrating the timeline of the segments. Bottom: table containing the total lengths of those segments. First three columns: total sufficient quality uptimes in each of the detectors. Following three columns: total lengths of the pairwise intersections of these segments, and in the last column, the total length of their intersection over all three detectors.

Contemporary GW detectors are very complicated tools and require nearly continuous maintenance, leading to a lot of downtimes, as seen in Fig. 6.1. In fact, in Sec. 5.7, only taking segments where data in both H1 and L1 was available in sufficient quality left us with less than two thirds of the O3b observing run. It would be beneficial for a search to also include V1 data; at the same time, this would likely leave us with an even smaller part of the observing run’s duration.

A possible solution is to train multiple search algorithms to account for the up-/downtimes of different combination of detectors. In this thesis, we have covered the situation of a single detector in Ch. 4, and due to the fundamental symmetries of the problem, the solution is applicable in any particular detector with a similar noise characteristic. While no noise transients were included, it is likely that following the methods of Ch. 5 one could incorporate them into the single-detector searches.

The two-detector situation has been covered in Ch. 5 in the H1-L1 case, including treatment of noise transients. In the case of multiple detectors, their relative geographic position and orientation does affect the training data distribution, and therefore a search would need to be retrained to achieve optimal sensitivity during downtimes of H1 or L1. A coincident search would also likely perform well in this case [352].

In a three-detector case, it is also expected that the architecture of Tab. 5.5, with some modifications, would perform well. As more detectors join the network, such as KAGRA in Japan, all possible situations can be covered this way. While coincident searches are easier to extend to more detectors, fully coherent searches are more sensitive in networks with more than two detectors [384–386].

At the same time, this method is inefficient as the number of CNNs required to cover all cases over n detectors is $2^n - 1$ (when one algorithm is sufficient to cover all single-detector cases, it is $2^n - n$ instead). In addition, the ranking statistics produced by separately trained networks are likely to be mutually incompatible. It is thus highly desirable to find ML-based algorithms which are more flexible in this regard as well as reproduce the sensitivity of the algorithms described in this thesis and extend their comparative performance to matched-filter based searches to higher numbers of detectors.

Furthermore, throughout this thesis, we have focused merely on comparable-mass BBH mergers with masses in the $[10M_\odot, 50M_\odot]$ mass range. Staying in the realm of LIGO-Virgo-KAGRA binaries, it is critical to also cover masses outside this range, as evidenced by the results of Sec. 5.7, as well as BNS and BHNS mergers.

This presents us with the issue of very different inspiral times; while those we investigated are on the order of 1 second, extending the range to $[7M_\odot, 50M_\odot]$ already allows inspirals up to a length of 20 seconds. In the case of BNS mergers, the source may spend more than a minute in a detector’s sensitive band, and any conceivable method of extending the input window length of a CNN without considerable downsampling leads to a quick rise in the model complexity.

On this scale, the problem may be resolved using a piece-wise downsampling, retaining the higher-frequency content at the end of the input window [279, 352], or simply using a shorter input; this way, however, a considerable part of the waveform’s SNR is not seen by the search, limiting its sensitivity. Looking further into the future at processing of LISA data, it is necessary to develop algorithms which search for waveforms spending up to months in the sensitive band of a detector. In addition, it is likely to encounter overlapping signals in this case.

A potential solution here lies in the use of recurrent neural networks, in particular long short-term memory networks [387], which keep an internal state. When applied sequentially to time series slices or individual elements, they may be capable of effectively searching for signals longer than the input window, or reject long noise transients effectively [171, 176, 183, 187, 388, 389].

Another issue in the usage of CNNs is a reduced understanding of the model's inner workings in comparison to matched-filter based pipelines. Along with DNN-based architectures, an associated trend is *interpretability* of those architectures [390–395]. An important direction in future research will be to make sure ML-based models are interpretable rather than black boxes [396, 397].

Throughout the time spent working on these projects, I have also taught in a course focused on ML for physicists completed by student projects, and seen many interesting applications to real physics problems by the students (mostly with no relation to GW research, and rather frequently in applied optics). The projects covered in this dissertation as well as these student projects and many others demonstrate that ML methods are a very useful part of a physicist's toolbox.

In conclusion, ML applications to physical problems are expected to become a standard part of a physicist's toolbox in the near future. There is great potential for their use in GW astronomy which is being exploited by more and more researchers and it is an exciting time of this rapidly growing field. Perhaps there will soon be ML-based online searches in use by the LIGO-Virgo-KAGRA collaboration, leading to joint analyses using both ML and matched-filter based methods, exploiting the strengths of both approaches.

Appendix A

Experiment with an extended mass range

In Sec. 5.5, we have trained a set of six networks to identify BBH waveforms in the MLGWSC-1 test data, optimized on the real detector noise of dataset 4 (see Sec. 5.1.4). As the architecture is best suited to detecting short signals, we have used component masses $m_1, m_2 \in [10M_\odot, 50M_\odot]$ for the training data waveforms, with a length on the order of 1 second. Therefore the waveforms with $m_2 \in [7M_\odot, 10M_\odot]$ and a maximum length ≈ 20 s are not represented in the training dataset while contained in test datasets 2-4.

An identical experiment has been performed with a single difference: the component masses for the training dataset are drawn from a uniform distribution over $[7M_\odot, 50M_\odot]$. The evolution of the sensitivity over datasets 3 and 4 at $\mathcal{F} = 1 \text{ month}^{-1}$ is shown in Fig. A.1 and is a direct equivalent of Fig. 5.6. Again, we decided to focus on the global maximum of each curve in the bottom panel, i.e. on dataset 4, and denote the network states in the format `E<run number>/<four-digit epoch number>`.

In two runs, this maximum exceeds 1300 Mpc (the higher being the state E4/0036, in one it is $\doteq 950$ Mpc, and in the remaining three it exceeds 1200 Mpc but remains below 1300 Mpc. In addition, we evaluate the E4/0036 state on the datasets used for the MLGWSC-1 results, and display the sensitivity curves in Fig. A.2. While this is a significant improvement over our original submission, the corrected results of Sec. 5.1.4, namely the R1/0021 search, seem to perform slightly better in the framework of the MLGWSC-1.

Finally, we applied the six networks selected by the maxima of the sensitivity on dataset 4 at a FAR of 1 per month to O3b data, taking the same steps as in Sec. 5.7. The results are shown in Tab. A.1. In comparison to the analysis of Tab. 5.6, we see an expected increase of events with component masses in the training range, which is now

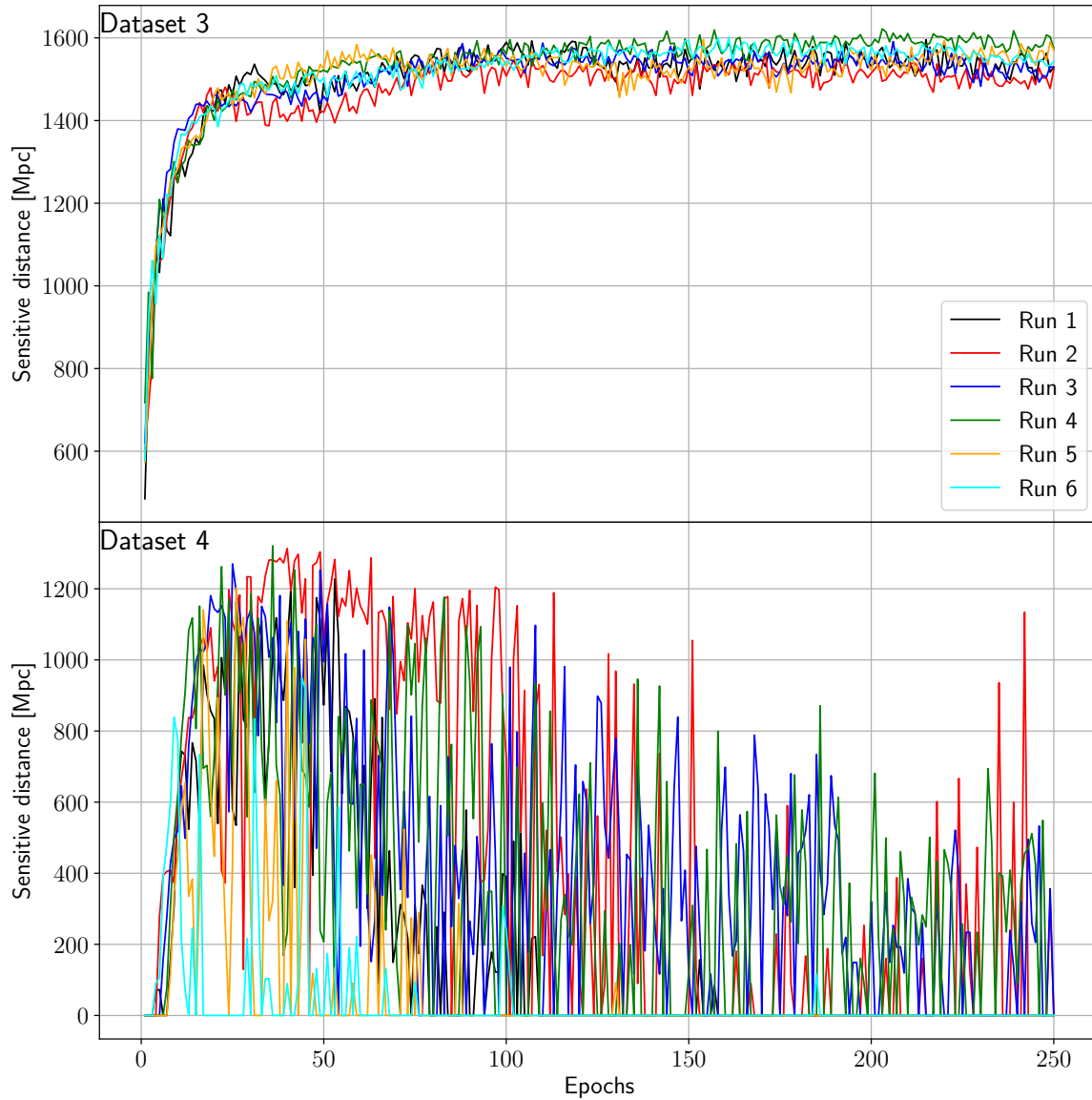


FIGURE A.1: Evolution of the sensitive distance during training for a set of 6 training runs. The sensitive distance in the top and bottom panels is evaluated on datasets 3 and 4, respectively, at $\mathcal{F} = 1$ per month.

Event name	ρ_{MF}	E1/0053	E2/0040	E3/0025	E4/0036	E5/0026	E6/0044
		\mathcal{F} [month ⁻¹]					
GW200224_222234	20.0	0.6	0.0	0.0	0.0	0.0	3.1
GW200311_115853	17.8	0.6	0.9	0.0	0.0	0.0	10.1
GW200225_060421	12.5	0.6	0.0	0.0	0.0	0.0	5.4
GW191215_223052	11.2	1.6	0.0	0.3	0.0	0.3	20.5
GW200208_130117	10.8	6.9	4.4	1.6	5.7	2.2	43.5
GW200219_094415	10.7	24.3	3.8	9.4	2.8	6.0	60.5
GW200209_085452	9.6	8.2	0.9	3.1	0.0	5.4	17.6
GW191204_110529	8.8	6.3	0.9	12.0	1.6	7.2	35.6
GW200306_093714	7.8	812	-	806	486	-	-
GW200308_173609	7.1	-	-	-	-	-	-
GW191204_171526	17.5	1.9	0.0	4.1	2.2	0.3	11.7
GW191129_134029	13.1	188	86.6	-	-	-	657
GW191222_033537	12.5	8.8	0.9	0.3	0.0	0.0	17.6
GW200202_154313	10.8	-	-	-	-	-	-
GW200128_022011	10.6	3.8	0.9	6.3	3.1	20.2	34.6
GW191230_180458	10.4	25.8	8.2	19.2	6.9	40.0	59.8
GW200316_215756	10.3	-	-	-	-	-	-
GW191105_143521	9.7	204	49.1	660	283	816	395
GW191127_050227	9.2	3.8	1.6	12.6	1.3	14.5	45.7
GW191103_012549	8.9	-	-	-	-	-	-
GW200220_124850	8.5	66.1	52.9	488	123	47.2	173
GW191126_115259	8.3	-	-	-	-	-	-
GW200216_220804	8.1	-	-	-	650	-	-
GW191113_071753	7.9	728	335	289	593	115	838
GW200208_222617	7.4	54.8	37.2	157	441	155	123
GW200322_091133	6.0	-	-	-	-	-	-
GW191109_010717	17.3	5.7	1.6	0.0	0.0	0.0	10.1
GW200115_042309	11.3	-	-	-	-	-	-
GW191219_163120	9.1	-	-	-	-	-	-
GW200210_092254	8.4	-	-	-	-	-	-
GW200220_061928	7.2	-	-	-	-	-	-

TABLE A.1: List of O3b events and their identification by the 6 searches at the peak of each curve in the bottom panel of Fig. A.1. Only events from the GWTC-3-confident catalog [72] are included. Events which were not recovered by the given search are marked by a hyphen. The accompanying SNR values come from the catalog’s parameter inference pipeline based on Bilby [78, 79, 382]. In addition, 4 events were omitted due to insufficient data quality or availability.

The events are split into three sections by horizontal lines. The top section contains events whose 90% credible intervals for both component masses lie fully in the interval $[7M_{\odot}, 50M_{\odot}]$ used to generate the training data. For events in the second section, at least one does not fulfill this but both intervals lie at least partly in the training range, and in the third section, at least one of the intervals lies fully outside the training range. Within each section, the events are sorted by their SNRs in a descending order. The credible intervals come from the parameter estimation supplied by GWOSC [383].

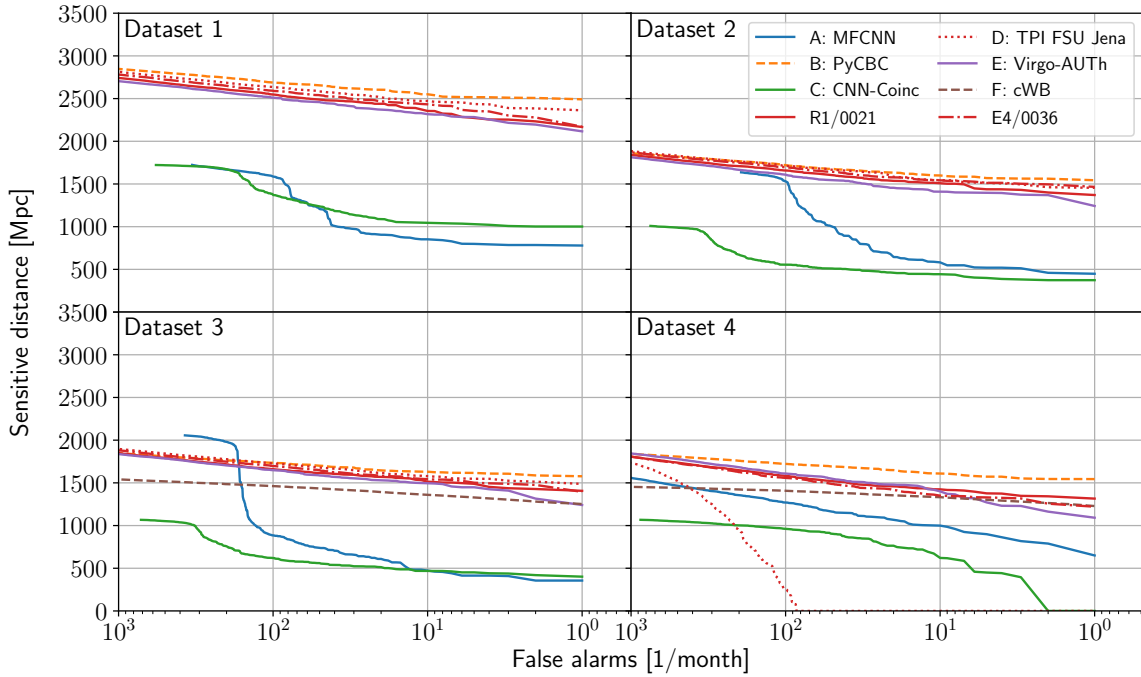


FIGURE A.2: Sensitivity curves of all original submissions over the 4 test datasets used to evaluate the MLGWSC-1. The R1/0021 algorithm replaces our original submission, which is represented by a dotted line instead, and the E4/0036 search is added as the dash-dotted line.

$[7M_{\odot}, 50M_{\odot}]$: 10 instead of 9 events with 90% credible intervals of both masses in the training range, 16 instead of 11 with both credible intervals at least partially in the training range but at least one not fully, and 5 instead of 11 with at least one credible interval fully outside the training range. However, we do not observe a significant improvement in the FAR assigned to either of the events in comparison to Tab. 5.6.

Appendix B

Segment keys

This appendix briefly lists the keys of real noise segments used in Ch. 5. Namely, Tab. B.1 lists keys of data used to slice into noise for the training data, and Tab. B.2 keys of data used to generate the test datasets used for the algorithm development and in the MLGWSC-1 evaluation. The overlap of these two lists is highlighted in bold.

Finally, Tab. B.3 lists keys of data used to generate the dataset for final validation as described in Sec. 5.6. It does not overlap with either of the previous two and thus there is no need for highlighting selected keys.

1238758280	1239087144	1239153852	1240944877	1241569759	1241597289
1242107494	1242258122	1242610603	1242787561	1242857540	1243393030
1243630285	1243831494	1243964965	1244839875	1244866703	1245047190
1245191877	1245332582	1245451429	1245742090	1245761268	1246048419
1246591138	1246757180	1246780061	1247517902	1247555609	1247664213
1248174787	1248435939	1249629768	1249781664	1250041152	1250152309
1250397388	1250430837	1250553015	1250709596	1250746105	1251862303
1252541512	1252694121	1252720941	1252808733	1253156172	1253717896
1253741480	1253853775				

TABLE B.1: Keys of 50 segments used to slice into training data. Printed in bold are those which overlap with Tab. B.2.

1238205077	1238400368	1238546548	1238561176	1238645908	1238677592
1238731472	1238758280	1238805513	1238819847	1238834634	1238918966
1238996844	1239060175	1239069672	1239087144	1239153852	1239168627
1239213619	1239449416	1239672094	1239795022	1240266411	1240285237
1240305829	1240360864	1240374893	1240398856	1240554257	1240594566
1240624416	1240658946	1240704124	1240712035	1240750853	1240764595
1240797030	1240821143	1240872265	1240919432	1240944877	1240961313
1241093496	1241123882	1241204835	1241243173	1241302064	1241310401
1241417186	1241508299	1241546152	1241569759	1241597289	1241619262
1241651790	1241689644	1241739106	1241758890	1241833954	1241857376
1241939956	1242071794	1242107494	1242124680	1242189936	1242242537
1242258122	1242279531	1242326536	1242346161	1242401394	1242442982
1242451141	1242459872	1242522327	1242535516	1242558506	1242567217
1242576179	1242610603	1242645739	1242692672	1242773677	1242787561
1242857540	1242877872	1242966830	1242997805	1243053828	1243078770
1243117125	1243162956	1243217565	1243254523	1243281488	1243330888
1243345012	1243393030	1243449741	1243518145	1243541014	1243560610
1243586469	1243630285	1243645650	1243673479	1243736093	1243754588
1243779090	1243823546	1243831494	1243888254	1243915837	1243947945
1243964965	1243993765	1244056996	1244067109	1244082986	1244106430
1244259779	1244331130	1244349588	1244367288	1244421366	1244471454
1244486004	1244499334	1244522898	1244543819	1244593951	

TABLE B.2: Keys of 131 segments used in the test datasets used for development and to evaluate the MLGWSC-1. Printed in bold are those which overlap with Tab. B.1.

1244608795	1244646462	1244654715	1244704480	1244772092	1244936272
1244986313	1245075223	1245091570	1245130318	1245200586	1245256976
1245308619	1245342373	1245464815	1245475577	1245569309	1245579592
1245653511	1245781152	1245828293	1245853320	1245892667	1245910462
1245967232	1246084506	1246170217	1246209961	1246239228	1246261454
1246286035	1246328016	1246378432	1246453148	1246487234	1246509724
1246527239	1246561826	1246631092	1246681818	1246841314	1246865565
1247260178	1247296003	1247359793	1247388908	1247401589	1247442444
1247460978	1247470158	1247583307	1247639142	1247694238	1247741180
1247750009	1247773736	1247810072	1247874995	1248112051	1248132105
1248200542	1248227246	1248242646	1248257816	1248283368	1248301521
1248331543	1248361429	1248381203	1248476855	1248511438	1248593325
1248683713	1248743587	1248798693	1248813258	1248834454	1248959459
1248991663	1249000658	1249093446	1249126630	1249163431	1249196261
1249216161	1249276049	1249323317	1249345682	1249367294	1249378088
1249399139	1249416034	1249428701	1249467944	1249497203	1249557842
1249584314	1249615368	1249663752	1249689290	1249716729	1249871829
1250031076	1250058904	1250108107	1250136805	1250196966	1250228595
1250242200	1250302734	1250473133	1250506415	1250617985	1250643575
1250656435	1250675600	1250799366	1250821662	1250848841	1250866922
1250916949	1250997653	1251010542	1251043190	1251176536	1251213336
1251227166	1251242198	1251261339	1251292071	1251332604	1251353035
1251420609	1251454669	1251523468	1251591106	1251646225	1251678305

TABLE B.3: Keys of 136 segments used in the test dataset used for the final evaluation.

Bibliography

- [1] Albert Einstein. “The Field Equations of Gravitation”. In: *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)* 1915 (1915), pp. 844–847.
- [2] Joel M. Weisberg et al. “Gravitational Waves from an Orbiting Pulsar”. In: *Scientific American* 245.4 (Oct. 1981), pp. 74–82. DOI: 10.1038/scientificamerican1081-74. URL: <https://doi.org/10.1038/scientificamerican1081-74>.
- [3] Joel M. Weisberg et al. “Relativistic binary pulsar B1913+16: Thirty years of observations and analysis”. In: *ASP Conf. Ser.* 328 (2005), p. 25. arXiv: astro-ph/0407149.
- [4] Nobel Prize Outreach AB 2023. *The Nobel Prize in Physics 1993*. NobelPrize.org. 2023. URL: <https://www.nobelprize.org/prizes/physics/1993/press-release/> (visited on 05/12/2023).
- [5] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116.6 (2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. arXiv: 1602.03837 [gr-qc].
- [6] B. P. Abbott et al. “Properties of the Binary Black Hole Merger GW150914”. In: *Phys. Rev. Lett.* 116 (24 2016), p. 241102. DOI: 10.1103/PhysRevLett.116.241102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.116.241102>.
- [7] Alex Abramovici et al. “LIGO: The Laser Interferometer Gravitational-Wave Observatory”. In: *Science* 256.5055 (1992), pp. 325–333. DOI: 10.1126/science.256.5055.325. eprint: <https://www.science.org/doi/pdf/10.1126/science.256.5055.325>. URL: <https://www.science.org/doi/abs/10.1126/science.256.5055.325>.
- [8] B P Abbott et al. “LIGO: the Laser Interferometer Gravitational-Wave Observatory”. In: *Reports on Progress in Physics* 72.7 (2009), p. 076901. DOI: 10.1088/0034-4885/72/7/076901. URL: <https://dx.doi.org/10.1088/0034-4885/72/7/076901>.

- [9] B. P. Abbott et al. “GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral”. In: *Phys. Rev. Lett.* 119 (16 2017), p. 161101. DOI: 10.1103/PhysRevLett.119.161101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.119.161101>.
- [10] B. P. Abbott et al. “Properties of the Binary Neutron Star Merger GW170817”. In: *Phys. Rev. X* 9 (1 2019), p. 011001. DOI: 10.1103/PhysRevX.9.011001. URL: <https://link.aps.org/doi/10.1103/PhysRevX.9.011001>.
- [11] R. Abbott et al. “GW190521: A Binary Black Hole Merger with a Total Mass of $150M_{\odot}$ ”. In: *Phys. Rev. Lett.* 125.10 (2020), p. 101102. DOI: 10.1103/PhysRevLett.125.101102. arXiv: 2009.01075 [gr-qc].
- [12] R. Gamba et al. “GW190521 as a dynamical capture of two nonspinning black holes”. In: *Nature Astronomy* 7.1 (Nov. 2022), pp. 11–17. DOI: 10.1038/s41550-022-01813-w. URL: <https://doi.org/10.1038/s41550-022-01813-w>.
- [13] V. Savchenko et al. “INTEGRAL Detection of the First Prompt Gamma-Ray Signal Coincident with the Gravitational-wave Event GW170817”. In: *The Astrophysical Journal Letters* 848.2 (2017), p. L15. DOI: 10.3847/2041-8213/aa8f94. URL: <https://dx.doi.org/10.3847/2041-8213/aa8f94>.
- [14] A. Goldstein et al. “An Ordinary Short Gamma-Ray Burst with Extraordinary Implications: Fermi-GBM Detection of GRB 170817A”. In: *The Astrophysical Journal Letters* 848.2 (2017), p. L14. DOI: 10.3847/2041-8213/aa8f41. URL: <https://dx.doi.org/10.3847/2041-8213/aa8f41>.
- [15] B. P. Abbott et al. “Multi-messenger Observations of a Binary Neutron Star Merger*[†]”. In: *The Astrophysical Journal Letters* 848.2 (2017), p. L12. DOI: 10.3847/2041-8213/aa91c9. URL: <https://dx.doi.org/10.3847/2041-8213/aa91c9>.
- [16] B. P. Abbott et al. “Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A”. In: *The Astrophysical Journal Letters* 848.2 (2017), p. L13. DOI: 10.3847/2041-8213/aa920c. URL: <https://dx.doi.org/10.3847/2041-8213/aa920c>.
- [17] A. M. Ghez et al. “The accelerations of stars orbiting the Milky Way’s central black hole”. In: *Nature* 407.6802 (Sept. 2000), pp. 349–351. DOI: 10.1038/35030032. URL: <https://doi.org/10.1038/35030032>.

- [18] Florian Peißker et al. “S62 and S4711: Indications of a Population of Faint Fast-moving Stars inside the S2 Orbit—S4711 on a 7.6 yr Orbit around Sgr A*[†]”. In: *The Astrophysical Journal* 899.1 (2020), p. 50. DOI: 10.3847/1538-4357/ab9c1c. URL: <https://dx.doi.org/10.3847/1538-4357/ab9c1c>.
- [19] Kazunori Akiyama et al. “First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole”. In: *Astrophys. J. Lett.* 875 (2019), p. L1. DOI: 10.3847/2041-8213/ab0ec7. arXiv: 1906.11238 [astro-ph.GA].
- [20] Kazunori Akiyama et al. “First Sagittarius A* Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole in the Center of the Milky Way”. In: *Astrophys. J. Lett.* 930.2 (2022), p. L12. DOI: 10.3847/2041-8213/ac6674.
- [21] Jenny E. Greene et al. “Intermediate-Mass Black Holes”. In: *Annual Review of Astronomy and Astrophysics* 58.1 (2020), pp. 257–312. DOI: 10.1146/annurev-astro-032620-021835. eprint: <https://doi.org/10.1146/annurev-astro-032620-021835>. URL: <https://doi.org/10.1146/annurev-astro-032620-021835>.
- [22] B. P. Abbott et al. “Binary Black Hole Population Properties Inferred from the First and Second Observing Runs of Advanced LIGO and Advanced Virgo”. In: *The Astrophysical Journal Letters* 882.2 (2019), p. L24. DOI: 10.3847/2041-8213/ab3800. URL: <https://dx.doi.org/10.3847/2041-8213/ab3800>.
- [23] R. Abbott et al. “Population of Merging Compact Binaries Inferred Using Gravitational Waves through GWTC-3”. In: *Phys. Rev. X* 13 (1 2023), p. 011048. DOI: 10.1103/PhysRevX.13.011048. URL: <https://link.aps.org/doi/10.1103/PhysRevX.13.011048>.
- [24] R. Abbott et al. “Observation of Gravitational Waves from Two Neutron Star–Black Hole Coalescences”. In: *The Astrophysical Journal Letters* 915.1 (2021), p. L5. DOI: 10.3847/2041-8213/ac082e. URL: <https://dx.doi.org/10.3847/2041-8213/ac082e>.
- [25] Matteo Breschi. “Inferring the equation of state with multi-messenger signals from binary neutron star mergers”. PhD thesis. Jena U., 2023.
- [26] Tim Dietrich et al. “Matter imprints in waveform models for neutron star binaries: Tidal and self-spin effects”. In: *Phys. Rev. D* 99 (2 2019), p. 024029. DOI: 10.1103/PhysRevD.99.024029. URL: <https://link.aps.org/doi/10.1103/PhysRevD.99.024029>.

- [27] David Eichler et al. “Nucleosynthesis, neutrino bursts and γ -rays from coalescing neutron stars”. In: *Nature* 340.6229 (July 1989), pp. 126–128. DOI: 10.1038/340126a0. URL: <https://doi.org/10.1038/340126a0>.
- [28] C. Freiburghaus et al. “r-Process in Neutron Star Mergers”. In: *The Astrophysical Journal* 525.2 (1999), p. L121. DOI: 10.1086/312343. URL: <https://dx.doi.org/10.1086/312343>.
- [29] Daniel Kasen et al. “Origin of the heavy elements in binary neutron-star mergers from a gravitational-wave event”. In: *Nature* 551.7678 (Oct. 2017), pp. 80–84. DOI: 10.1038/nature24453. URL: <https://doi.org/10.1038/nature24453>.
- [30] James Liebert. “White Dwarf Stars”. In: *Annual Review of Astronomy and Astrophysics* 18.1 (1980), pp. 363–398. DOI: 10.1146/annurev.aa.18.090180.002051. eprint: <https://doi.org/10.1146/annurev.aa.18.090180.002051>. URL: <https://doi.org/10.1146/annurev.aa.18.090180.002051>.
- [31] Michela Mapelli. “Binary Black Hole Mergers: Formation and Populations”. In: *Frontiers in Astronomy and Space Sciences* 7 (2020). ISSN: 2296-987X. DOI: 10.3389/fspas.2020.00038. URL: <https://www.frontiersin.org/articles/10.3389/fspas.2020.00038>.
- [32] Ryan M. O’Leary et al. “Dynamical formation signatures of black hole binaries in the first detected mergers by LIGO”. In: *The Astrophysical Journal Letters* 824.1 (2016), p. L12. DOI: 10.3847/2041-8205/824/1/L12. URL: <https://dx.doi.org/10.3847/2041-8205/824/1/L12>.
- [33] Qingbo Chu et al. “Formation and evolution of binary neutron stars: mergers and their host galaxies”. In: *Monthly Notices of the Royal Astronomical Society* 509.2 (Oct. 2021), pp. 1557–1586. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2882. eprint: <https://academic.oup.com/mnras/article-pdf/509/2/1557/41157683/stab2882.pdf>. URL: <https://doi.org/10.1093/mnras/stab2882>.
- [34] B. P. Abbott et al. “Tests of General Relativity with GW150914”. In: *Phys. Rev. Lett.* 116 (22 2016), p. 221101. DOI: 10.1103/PhysRevLett.116.221101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.116.221101>.
- [35] B. P. Abbott et al. “Tests of General Relativity with GW170817”. In: *Phys. Rev. Lett.* 123 (1 2019), p. 011102. DOI: 10.1103/PhysRevLett.123.011102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.123.011102>.

- [36] B. P. Abbott et al. “Tests of general relativity with the binary black hole signals from the LIGO-Virgo catalog GWTC-1”. In: *Phys. Rev. D* 100 (10 2019), p. 104036. DOI: 10.1103/PhysRevD.100.104036. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.104036>.
- [37] Konstantin N Yakunin et al. “Gravitational waves from core collapse supernovae”. In: *Classical and Quantum Gravity* 27.19 (2010), p. 194005. DOI: 10.1088/0264-9381/27/19/194005. URL: <https://dx.doi.org/10.1088/0264-9381/27/19/194005>.
- [38] Ernazar Abdikamalov et al. “Gravitational Waves from Core-Collapse Supernovae”. In: (2020). Ed. by Cosimo Bambi et al., pp. 1–37. DOI: 10.1007/978-981-15-4702-7_21-1. URL: https://doi.org/10.1007/978-981-15-4702-7_21-1.
- [39] S. Bonazzola et al. “Gravitational waves from pulsars: Emission by the magnetic field induced distortion”. In: *Astron. Astrophys.* 312 (1996), p. 675. arXiv: [astro-ph/9602107](https://arxiv.org/abs/astro-ph/9602107).
- [40] Magdalena Sieniawska et al. “Continuous Gravitational Waves from Neutron Stars: Current Status and Prospects”. In: *Universe* 5.11 (2019). ISSN: 2218-1997. DOI: 10.3390/universe5110217. URL: <https://www.mdpi.com/2218-1997/5/11/217>.
- [41] R. Abbott et al. “Constraints from LIGO O3 Data on Gravitational-wave Emission Due to R-modes in the Glitching Pulsar PSR J0537–6910”. In: *The Astrophysical Journal* 922.1 (2021), p. 71. DOI: 10.3847/1538-4357/ac0d52. URL: <https://dx.doi.org/10.3847/1538-4357/ac0d52>.
- [42] Keith Riles. “Searches for continuous-wave gravitational radiation”. In: *Living Reviews in Relativity* 26.1 (Apr. 2023). DOI: 10.1007/s41114-023-00044-3. URL: <https://doi.org/10.1007/s41114-023-00044-3>.
- [43] P. B. Covas et al. “First All-Sky Search for Continuous Gravitational-Wave Signals from Unknown Neutron Stars in Binary Systems Using Advanced LIGO Data”. In: *Phys. Rev. Lett.* 124 (19 2020), p. 191102. DOI: 10.1103/PhysRevLett.124.191102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.124.191102>.

- [44] Leor Barack et al. “Black holes, gravitational waves and fundamental physics: a roadmap”. In: *Classical and Quantum Gravity* 36.14 (2019), p. 143001. DOI: 10.1088/1361-6382/ab0587. URL: <https://dx.doi.org/10.1088/1361-6382/ab0587>.
- [45] Luke Sellers et al. “Searching for Intelligent Life in Gravitational Wave Signals Part I: Present Capabilities and Future Horizons”. In: (Dec. 2022). arXiv: 2212.02065 [astro-ph.IM].
- [46] J. Weber. “Gravitational Radiation”. In: *Phys. Rev. Lett.* 18 (13 1967), pp. 498–501. DOI: 10.1103/PhysRevLett.18.498. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.18.498>.
- [47] J. Weber. “Evidence for Discovery of Gravitational Radiation”. In: *Phys. Rev. Lett.* 22 (24 1969), pp. 1320–1324. DOI: 10.1103/PhysRevLett.22.1320. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.22.1320>.
- [48] A de Waard et al. “MiniGRAIL, the first spherical detector”. In: *Classical and Quantum Gravity* 20.10 (2003), S143. DOI: 10.1088/0264-9381/20/10/317. URL: <https://dx.doi.org/10.1088/0264-9381/20/10/317>.
- [49] Andrea N Lommen. “Pulsar timing arrays: the promise of gravitational wave detection”. In: *Reports on Progress in Physics* 78.12 (2015), p. 124901. DOI: 10.1088/0034-4885/78/12/124901. URL: <https://dx.doi.org/10.1088/0034-4885/78/12/124901>.
- [50] Benjamin P Abbott et al. “A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals”. In: *Class. Quant. Grav.* 37.5 (2020), p. 055002. DOI: 10.1088/1361-6382/ab685e. arXiv: 1908.11170 [gr-qc].
- [51] Bruce Allen et al. “FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries”. In: *Phys. Rev. D* 85 (12 2012), p. 122006. DOI: 10.1103/PhysRevD.85.122006. URL: <https://link.aps.org/doi/10.1103/PhysRevD.85.122006>.
- [52] Samantha A Usman et al. “The PyCBC search for gravitational waves from compact binary coalescence”. In: *Classical and Quantum Gravity* 33.21 (2016), p. 215004. DOI: 10.1088/0264-9381/33/21/215004. URL: <https://dx.doi.org/10.1088/0264-9381/33/21/215004>.

- [53] Bruce Allen. “ χ^2 time-frequency discriminator for gravitational wave detection”. In: *Phys. Rev. D* 71 (6 2005), p. 062001. DOI: 10.1103/PhysRevD.71.062001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.71.062001>.
- [54] Alexander H. Nitz et al. “Detecting binary compact-object mergers with gravitational waves: Understanding and Improving the sensitivity of the PyCBC search”. In: *Astrophys. J.* 849.2 (2017), p. 118. DOI: 10.3847/1538-4357/aa8f50. arXiv: 1705.01513 [gr-qc].
- [55] Tito Dal Canton et al. “Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors”. In: *Phys. Rev. D* 90 (8 2014), p. 082004. DOI: 10.1103/PhysRevD.90.082004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.90.082004>.
- [56] Alexander H. Nitz et al. “Rapid detection of gravitational waves from compact binary mergers with PyCBC Live”. In: *Phys. Rev. D* 98 (2 2018), p. 024050. DOI: 10.1103/PhysRevD.98.024050. URL: <https://link.aps.org/doi/10.1103/PhysRevD.98.024050>.
- [57] Tito Dal Canton et al. “Real-time Search for Compact Binary Mergers in Advanced LIGO and Virgo’s Third Observing Run Using PyCBC Live”. In: *The Astrophysical Journal* 923.2 (2021), p. 254. DOI: 10.3847/1538-4357/ac2f9a. URL: <https://dx.doi.org/10.3847/1538-4357/ac2f9a>.
- [58] Cody Messick et al. “Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data”. In: *Phys. Rev. D* 95 (4 2017), p. 042001. DOI: 10.1103/PhysRevD.95.042001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.95.042001>.
- [59] Stephen Privitera et al. “Improving the sensitivity of a search for coalescing binary black holes with nonprecessing spins in gravitational wave data”. In: *Phys. Rev. D* 89 (2 2014), p. 024003. DOI: 10.1103/PhysRevD.89.024003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.89.024003>.
- [60] Kipp Cannon et al. “Interpolation in waveform space: Enhancing the accuracy of gravitational waveform families using numerical relativity”. In: *Phys. Rev. D* 87 (4 2013), p. 044008. DOI: 10.1103/PhysRevD.87.044008. URL: <https://link.aps.org/doi/10.1103/PhysRevD.87.044008>.

- [61] Kipp Cannon et al. “Method to estimate the significance of coincident gravitational-wave observations from compact binary coalescence”. In: *Phys. Rev. D* 88 (2 2013), p. 024025. DOI: 10.1103/PhysRevD.88.024025. URL: <https://link.aps.org/doi/10.1103/PhysRevD.88.024025>.
- [62] T Adams et al. “Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era”. In: *Classical and Quantum Gravity* 33.17 (2016), p. 175012. DOI: 10.1088/0264-9381/33/17/175012. URL: <https://dx.doi.org/10.1088/0264-9381/33/17/175012>.
- [63] F Aubin et al. “The MBTA pipeline for detecting compact binary coalescences in the third LIGO–Virgo observing run”. In: *Classical and Quantum Gravity* 38.9 (2021), p. 095004. DOI: 10.1088/1361-6382/abe913. URL: <https://dx.doi.org/10.1088/1361-6382/abe913>.
- [64] Marco Drago et al. “coherent WaveBurst, a pipeline for unmodeled gravitational-wave data analysis”. In: *SoftwareX* 14 (2021), p. 100678. ISSN: 2352-7110. DOI: <https://doi.org/10.1016/j.softx.2021.100678>. URL: <https://www.sciencedirect.com/science/article/pii/S2352711021000236>.
- [65] S. Klimenko et al. “Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors”. In: *Phys. Rev. D* 93 (4 2016), p. 042004. DOI: 10.1103/PhysRevD.93.042004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.93.042004>.
- [66] S Klimenko et al. “A coherent method for detection of gravitational wave bursts”. In: *Classical and Quantum Gravity* 25.11 (2008), p. 114029. DOI: 10.1088/0264-9381/25/11/114029. URL: <https://dx.doi.org/10.1088/0264-9381/25/11/114029>.
- [67] V Nacula et al. “Transient analysis with fast Wilson-Daubechies time-frequency transform”. In: *Journal of Physics: Conference Series* 363.1 (2012), p. 012032. DOI: 10.1088/1742-6596/363/1/012032. URL: <https://dx.doi.org/10.1088/1742-6596/363/1/012032>.
- [68] F. Salemi et al. “Wider look at the gravitational-wave transients from GWTC-1 using an unmodeled reconstruction method”. In: *Phys. Rev. D* 100 (4 2019), p. 042003. DOI: 10.1103/PhysRevD.100.042003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.042003>.

- [69] B. P. Abbott et al. “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs”. In: *Phys. Rev. X* 9 (3 2019), p. 031040. DOI: 10.1103/PhysRevX.9.031040. URL: <https://link.aps.org/doi/10.1103/PhysRevX.9.031040>.
- [70] R. Abbott et al. “GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo during the First Half of the Third Observing Run”. In: *Phys. Rev. X* 11 (2 2021), p. 021053. DOI: 10.1103/PhysRevX.11.021053. URL: <https://link.aps.org/doi/10.1103/PhysRevX.11.021053>.
- [71] R. Abbott et al. “GWTC-2.1: Deep Extended Catalog of Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run”. In: (Aug. 2021). arXiv: 2108.01045 [gr-qc].
- [72] R. Abbott et al. “GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run”. In: (Nov. 2021). arXiv: 2111.03606 [gr-qc].
- [73] Jing Luan et al. “Towards low-latency real-time detection of gravitational waves from compact binary coalescences in the era of advanced detectors”. In: *Phys. Rev. D* 85 (10 2012), p. 102002. DOI: 10.1103/PhysRevD.85.102002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.85.102002>.
- [74] Qi Chu. “Low-latency detection and localization of gravitational waves from compact binary coalescences”. English. PhD thesis. The University of Western Australia, 2017. DOI: 10.4225/23/5987feb0a789c.
- [75] Qi Chu et al. “SPIIR online coherent pipeline to search for gravitational waves from compact binary coalescences”. In: *Phys. Rev. D* 105 (2 2022), p. 024023. DOI: 10.1103/PhysRevD.105.024023. URL: <https://link.aps.org/doi/10.1103/PhysRevD.105.024023>.
- [76] Curt Cutler et al. “Gravitational waves from merging compact binaries: How accurately can one extract the binary’s parameters from the inspiral waveform?” In: *Phys. Rev. D* 49 (6 1994), pp. 2658–2697. DOI: 10.1103/PhysRevD.49.2658. URL: <https://link.aps.org/doi/10.1103/PhysRevD.49.2658>.
- [77] C. M. Biwer et al. “PyCBC Inference: A Python-based Parameter Estimation Toolkit for Compact Binary Coalescence Signals”. In: *Publications of the Astronomical Society of the Pacific* 131.996 (2019), p. 024503. DOI: 10.1088/1538-3873/aaef0b. URL: <https://dx.doi.org/10.1088/1538-3873/aaef0b>.

- [78] Gregory Ashton et al. “Bilby: A User-friendly Bayesian Inference Library for Gravitational-wave Astronomy”. In: *The Astrophysical Journal Supplement Series* 241.2 (2019), p. 27. DOI: 10.3847/1538-4365/ab06fc. URL: <https://dx.doi.org/10.3847/1538-4365/ab06fc>.
- [79] I M Romero-Shaw et al. “Bayesian inference for compact binary coalescences with bilby: validation and application to the first LIGO–Virgo gravitational-wave transient catalogue”. In: *Monthly Notices of the Royal Astronomical Society* 499.3 (Sept. 2020), pp. 3295–3319. ISSN: 0035-8711. DOI: 10.1093/mnras/staa2850. eprint: <https://academic.oup.com/mnras/article-pdf/499/3/3295/34052625/staa2850.pdf>. URL: <https://doi.org/10.1093/mnras/staa2850>.
- [80] G Ashton et al. “Bilby-MCMC: an MCMC sampler for gravitational-wave inference”. In: *Monthly Notices of the Royal Astronomical Society* 507.2 (Aug. 2021), pp. 2037–2051. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2236. eprint: <https://academic.oup.com/mnras/article-pdf/507/2/2037/44923607/stab2236.pdf>. URL: <https://doi.org/10.1093/mnras/stab2236>.
- [81] Homare Abe et al. “The Current Status and Future Prospects of KAGRA, the Large-Scale Cryogenic Gravitational Wave Telescope Built in the Kamioka Underground”. In: *Galaxies* 10.3 (2022), p. 63. DOI: 10.3390/galaxies10030063.
- [82] Alexander H. Nitz et al. “Search for Eccentric Binary Neutron Star Mergers in the First and Second Observing Runs of Advanced LIGO”. In: *The Astrophysical Journal* 890.1 (2020), p. 1. DOI: 10.3847/1538-4357/ab6611. URL: <https://dx.doi.org/10.3847/1538-4357/ab6611>.
- [83] Ian Harry et al. “Searching for gravitational waves from compact binaries with precessing spins”. In: *Phys. Rev. D* 94 (2 2016), p. 024012. DOI: 10.1103/PhysRevD.94.024012. URL: <https://link.aps.org/doi/10.1103/PhysRevD.94.024012>.
- [84] Ian Harry et al. “Searching for the full symphony of black hole binary mergers”. In: *Phys. Rev. D* 97 (2 2018), p. 023004. DOI: 10.1103/PhysRevD.97.023004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.97.023004>.
- [85] M Punturo et al. “The Einstein Telescope: a third-generation gravitational wave observatory”. In: *Classical and Quantum Gravity* 27.19 (2010), p. 194002. DOI: 10.1088/0264-9381/27/19/194002. URL: <https://dx.doi.org/10.1088/0264-9381/27/19/194002>.

- [86] Michele Maggiore et al. “Science case for the Einstein telescope”. In: *Journal of Cosmology and Astroparticle Physics* 2020.03 (2020), p. 050. DOI: 10.1088/1475-7516/2020/03/050. URL: <https://dx.doi.org/10.1088/1475-7516/2020/03/050>.
- [87] Matteo Breschi et al. “Kilohertz Gravitational Waves from Binary Neutron Star Mergers: Inference of Postmerger Signals with the Einstein Telescope”. In: (May 2022). arXiv: 2205.09979 [gr-qc].
- [88] B Sathyaprakash et al. “Scientific objectives of Einstein Telescope”. In: *Classical and Quantum Gravity* 29.12 (2012), p. 124013. DOI: 10.1088/0264-9381/29/12/124013. URL: <https://dx.doi.org/10.1088/0264-9381/29/12/124013>.
- [89] David Reitze et al. “Cosmic Explorer: The U.S. Contribution to Gravitational-Wave Astronomy beyond LIGO”. In: *Bull. Am. Astron. Soc.* 51.7 (2019), p. 035. arXiv: 1907.04833 [astro-ph.IM].
- [90] B P Abbott et al. “Exploring the sensitivity of next generation gravitational wave detectors”. In: *Classical and Quantum Gravity* 34.4 (2017), p. 044001. DOI: 10.1088/1361-6382/aa51f4. URL: <https://dx.doi.org/10.1088/1361-6382/aa51f4>.
- [91] Jonathan R. Gair et al. “Exploring intermediate and massive black-hole binaries with the Einstein Telescope”. In: *General Relativity and Gravitation* 43.2 (Oct. 2010), pp. 485–518. DOI: 10.1007/s10714-010-1104-3. URL: <https://doi.org/10.1007/s10714-010-1104-3>.
- [92] Jonathan R Gair et al. “Probing seed black holes using future gravitational-wave detectors”. In: *Classical and Quantum Gravity* 26.20 (2009), p. 204009. DOI: 10.1088/0264-9381/26/20/204009. URL: <https://dx.doi.org/10.1088/0264-9381/26/20/204009>.
- [93] Pau Amaro-Seoane et al. “Detection of IMBHs with ground-based gravitational wave observatories: A biography of a binary of black holes, from birth to death”. In: *The Astrophysical Journal* 722.2 (2010), p. 1197. DOI: 10.1088/0004-637X/722/2/1197. URL: <https://dx.doi.org/10.1088/0004-637X/722/2/1197>.
- [94] B. Sathyaprakash et al. “Scientific Potential of Einstein Telescope”. In: *46th Rencontres de Moriond on Gravitational Waves and Experimental Gravity*. Aug. 2011, pp. 127–136. arXiv: 1108.1423 [gr-qc].

- [95] Tania Regimbau. “The Quest for the Astrophysical Gravitational-Wave Background with Terrestrial Detectors”. In: *Symmetry* 14.2 (2022). ISSN: 2073-8994. DOI: 10.3390/sym14020270. URL: <https://www.mdpi.com/2073-8994/14/2/270>.
- [96] Tania Regimbau et al. “Mock data challenge for the Einstein Gravitational-Wave Telescope”. In: *Phys. Rev. D* 86 (12 2012), p. 122001. DOI: 10.1103/PhysRevD.86.122001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.86.122001>.
- [97] Chiara Caprini et al. “Cosmological backgrounds of gravitational waves”. In: *Classical and Quantum Gravity* 35.16 (2018), p. 163001. DOI: 10.1088/1361-6382/aac608. URL: <https://dx.doi.org/10.1088/1361-6382/aac608>.
- [98] Alessandra Buonanno et al. “Stochastic gravitational-wave background from cosmological supernovae”. In: *Phys. Rev. D* 72 (8 2005), p. 084001. DOI: 10.1103/PhysRevD.72.084001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.72.084001>.
- [99] R. Brustein et al. “Relic gravitational waves from string cosmology”. In: *Physics Letters B* 361.1 (1995), pp. 45–51. ISSN: 0370-2693. DOI: [https://doi.org/10.1016/0370-2693\(95\)01128-D](https://doi.org/10.1016/0370-2693(95)01128-D). URL: <https://www.sciencedirect.com/science/article/pii/037026939501128D>.
- [100] Ashish Sharma et al. “Searching for cosmological gravitational-wave backgrounds with third-generation detectors in the presence of an astrophysical foreground”. In: *Phys. Rev. D* 102 (6 2020), p. 063009. DOI: 10.1103/PhysRevD.102.063009. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.063009>.
- [101] L. P. Grishchuk. “Amplification of gravitational waves in an isotropic universe”. In: *Zh. Eksp. Teor. Fiz.* 67 (1974), pp. 825–838.
- [102] Xing-Jiang Zhu et al. “On the gravitational wave background from compact binary coalescences in the band of ground-based interferometers”. In: *Monthly Notices of the Royal Astronomical Society* 431.1 (Mar. 2013), pp. 882–899. ISSN: 0035-8711. DOI: 10.1093/mnras/stt207. eprint: <https://academic.oup.com/mnras/article-pdf/431/1/882/18243620/stt207.pdf>. URL: <https://doi.org/10.1093/mnras/stt207>.
- [103] Pau Amaro Seoane et al. “The Gravitational Universe”. In: (May 2013). arXiv: 1305.5720 [astro-ph.CO].

-
- [104] Max-Planck-Gesellschaft zur Förderung der Wissenschaften e.V. *LISA Consortium*. 2023. URL: <https://www.elisascience.org/> (visited on 05/14/2023).
- [105] National Aeronautics et al. *NASA LISA*. 2023. URL: <https://lisa.nasa.gov/> (visited on 05/14/2023).
- [106] Stanislav Babak et al. “Science with the space-based interferometer LISA. V. Extreme mass-ratio inspirals”. In: *Phys. Rev. D* 95 (10 2017), p. 103012. DOI: 10.1103/PhysRevD.95.103012. URL: <https://link.aps.org/doi/10.1103/PhysRevD.95.103012>.
- [107] Jan Harms et al. “Lunar Gravitational-wave Antenna”. In: *The Astrophysical Journal* 910.1 (2021), p. 1. DOI: 10.3847/1538-4357/abe5a7. URL: <https://dx.doi.org/10.3847/1538-4357/abe5a7>.
- [108] Pau Amaro-Seoane et al. “LION: laser interferometer on the moon”. In: *Classical and Quantum Gravity* 38.12 (2021), p. 125008. DOI: 10.1088/1361-6382/abf441. URL: <https://dx.doi.org/10.1088/1361-6382/abf441>.
- [109] Wikipedia contributors. *Timeline of machine learning — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Timeline_of_machine_learning&oldid=1152538889. [Online; accessed 14-May-2023]. 2023.
- [110] Ronan Collobert et al. “Torch: a modular machine learning software library”. In: (2002). URL: <http://infoscience.epfl.ch/record/82802>.
- [111] Kurt Hornik et al. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8). URL: <https://www.sciencedirect.com/science/article/pii/0893608089900208>.
- [112] Tin Kam Ho. “Random decision forests”. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1. 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.
- [113] Corinna Cortes et al. “Support-vector networks”. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. DOI: 10.1007/bf00994018. URL: <https://doi.org/10.1007/bf00994018>.

- [114] Pankaj Mehta et al. “A high-bias, low-variance introduction to Machine Learning for physicists”. In: *Physics Reports* 810 (2019). A high-bias, low-variance introduction to Machine Learning for physicists, pp. 1–124. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2019.03.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0370157319300766>.
- [115] Ian Goodfellow et al. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [116] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: (June 2014). arXiv: 1406.2661 [stat.ML].
- [117] Julia Dubenskaya et al. “Using a Conditional Generative Adversarial Network to Control the Statistical Characteristics of Generated Images for IACT Data Analysis”. In: *PoS DLCP2022* (2022), p. 004. DOI: 10.22323/1.429.0004. arXiv: 2211.15807 [astro-ph.IM].
- [118] Paul Lutkus et al. “Towards Designing and Exploiting Generative Networks for Neutrino Physics Experiments using Liquid Argon Time Projection Chambers”. In: *9th International Conference on Learning Representations*. Apr. 2022. arXiv: 2204.02496 [hep-ex].
- [119] David Silver et al. “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”. In: *arXiv e-prints*, arXiv:1712.01815 (Dec. 2017), arXiv:1712.01815. DOI: 10.48550/arXiv.1712.01815. arXiv: 1712.01815 [cs.AI].
- [120] David Silver et al. “Mastering the game of Go without human knowledge”. In: *Nature* 550.7676 (2017), pp. 354–359. ISSN: 1476-4687. DOI: 10.1038/nature24270. URL: <https://doi.org/10.1038/nature24270>.
- [121] Zhou Lu et al. “The Expressive Power of Neural Networks: A View from the Width”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6232–6240. ISBN: 9781510860964.
- [122] Alex Krizhevsky et al. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.

- [123] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [124] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [125] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [126] TensorFlow Developers. *TensorFlow*. Version v2.13.0-rc1. Specific TensorFlow versions can be found in the "Versions" list on the right side of this page.
See the full list of authors on GitHub. May 2023. DOI: 10.5281/zenodo.7987192. URL: <https://doi.org/10.5281/zenodo.7987192>.
- [127] Yangqing Jia et al. *Caffe: Convolutional Architecture for Fast Feature Embedding*. Orlando, Florida, USA, 2014. DOI: 10.1145/2647868.2654889. URL: <https://doi.org/10.1145/2647868.2654889>.
- [128] The Theano Development Team et al. “Theano: A Python framework for fast computation of mathematical expressions”. In: *arXiv preprint arXiv:1605.02688* (2016). URL: <https://arxiv.org/pdf/1605.02688>.
- [129] OpenAI. *ChatGPT: A Large-Scale Language Model*. <https://chat.openai.com>. Accessed on April 26, 2023. 2021.
- [130] Giuseppe Carleo et al. “Machine learning and the physical sciences”. In: *Rev. Mod. Phys.* 91 (4 2019), p. 045002. DOI: 10.1103/RevModPhys.91.045002. URL: <https://link.aps.org/doi/10.1103/RevModPhys.91.045002>.
- [131] Allison McCarn Deiana et al. “Applications and Techniques for Fast Machine Learning in Science”. In: *Frontiers in Big Data* 5 (2022). ISSN: 2624-909X. DOI: 10.3389/fdata.2022.787421. URL: <https://www.frontiersin.org/articles/10.3389/fdata.2022.787421>.
- [132] Philip Harris et al. “Physics Community Needs, Tools, and Resources for Machine Learning”. In: *Snowmass 2021*. Mar. 2022. arXiv: 2203.16255 [cs.LG].

- [133] S. Cocco et al. “Statistical physics and representations in real and artificial neural networks”. In: *Physica A: Statistical Mechanics and its Applications* 504 (2018). Lecture Notes of the 14th International Summer School on Fundamental Problems in Statistical Physics, pp. 45–76. ISSN: 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2017.11.153>. URL: <https://www.sciencedirect.com/science/article/pii/S0378437117312347>.
- [134] Dan Guest et al. “Deep Learning and Its Application to LHC Physics”. In: *Annual Review of Nuclear and Particle Science* 68.1 (2018), pp. 161–181. DOI: [10.1146/annurev-nucl-101917-021019](https://doi.org/10.1146/annurev-nucl-101917-021019). eprint: <https://doi.org/10.1146/annurev-nucl-101917-021019>. URL: <https://doi.org/10.1146/annurev-nucl-101917-021019>.
- [135] Georgia Karagiorgi et al. “Machine learning in the search for new fundamental physics”. In: *Nature Reviews Physics* 4.6 (May 2022), pp. 399–412. DOI: [10.1038/s42254-022-00455-1](https://doi.org/10.1038/s42254-022-00455-1). URL: <https://doi.org/10.1038/s42254-022-00455-1>.
- [136] Fernanda Psihas et al. “A review on machine learning for neutrino experiments”. In: *International Journal of Modern Physics A* 35.33 (2020), p. 2043005. DOI: [10.1142/S0217751X20430058](https://doi.org/10.1142/S0217751X20430058). eprint: <https://doi.org/10.1142/S0217751X20430058>. URL: <https://doi.org/10.1142/S0217751X20430058>.
- [137] Fabian Ruehle. “Data science applications to string theory”. In: *Physics Reports* 839 (2020). Data science applications to string theory, pp. 1–117. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2019.09.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0370157319303072>.
- [138] Murat Abdughani et al. “Supervised Deep Learning in High Energy Phenomenology: a Mini Review*”. In: *Communications in Theoretical Physics* 71.8 (2019), p. 955. DOI: [10.1088/0253-6102/71/8/955](https://dx.doi.org/10.1088/0253-6102/71/8/955). URL: <https://dx.doi.org/10.1088/0253-6102/71/8/955>.
- [139] Andrew J. Larkoski et al. “Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning”. In: *Physics Reports* 841 (2020). Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning, pp. 1–63. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2019.11.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0370157319303643>.

- [140] Steven Farrell et al. “Novel deep learning methods for track reconstruction”. In: *4th International Workshop Connecting The Dots 2018*. Oct. 2018. arXiv: 1810.06111 [hep-ex].
- [141] P. Baldi et al. “Searching for exotic particles in high-energy physics with deep learning”. In: *Nature Communications* 5.1 (July 2014). DOI: 10.1038/ncomms5308. URL: <https://doi.org/10.1038/ncomms5308>.
- [142] A. Aurisano et al. “A convolutional neural network neutrino event classifier”. In: *Journal of Instrumentation* 11.09 (2016), P09001. DOI: 10.1088/1748-0221/11/09/P09001. URL: <https://dx.doi.org/10.1088/1748-0221/11/09/P09001>.
- [143] J. Duarte et al. “Fast inference of deep neural networks in FPGAs for particle physics”. In: *Journal of Instrumentation* 13.07 (2018), P07027. DOI: 10.1088/1748-0221/13/07/P07027. URL: <https://dx.doi.org/10.1088/1748-0221/13/07/P07027>.
- [144] Daniel Guest et al. “Jet flavor classification in high-energy physics with deep neural networks”. In: *Phys. Rev. D* 94 (11 2016), p. 112002. DOI: 10.1103/PhysRevD.94.112002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.94.112002>.
- [145] Jonathan Carifio et al. “Machine learning in the string landscape”. In: *Journal of High Energy Physics* 2017.9 (Sept. 2017). DOI: 10.1007/jhep09(2017)157. URL: [https://doi.org/10.1007/jhep09\(2017\)157](https://doi.org/10.1007/jhep09(2017)157).
- [146] Nicholas M. Ball et al. “Data Mining and Machine Learning in Astronomy”. In: *International Journal of Modern Physics D* 19.07 (2010), pp. 1049–1106. DOI: 10.1142/S0218271810017160. eprint: <https://doi.org/10.1142/S0218271810017160>. URL: <https://doi.org/10.1142/S0218271810017160>.
- [147] Matthew Szydagis et al. “A Review of Basic Energy Reconstruction Techniques in Liquid Xenon and Argon Detectors for Dark Matter and Neutrino Physics Using NEST”. In: *Instruments* 5.1 (2021). ISSN: 2410-390X. DOI: 10.3390/instruments5010013. URL: <https://www.mdpi.com/2410-390X/5/1/13>.
- [148] Andrew E. Firth et al. “Estimating photometric redshifts with artificial neural networks”. In: *Monthly Notices of the Royal Astronomical Society* 339.4 (Mar. 2003), pp. 1195–1202. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2003.06271.x. eprint: <https://academic.oup.com/mnras/article-pdf/339/4/>

- 1195/18192282/339-4-1195.pdf. URL: <https://doi.org/10.1046/j.1365-8711.2003.06271.x>.
- [149] Thomas J Armitage et al. “An application of machine learning techniques to galaxy cluster mass estimation using the MACSIS simulations”. In: *Monthly Notices of the Royal Astronomical Society* 484.2 (Jan. 2019), pp. 1526–1537. ISSN: 0035-8711. DOI: 10.1093/mnras/stz039. eprint: <https://academic.oup.com/mnras/article-pdf/484/2/1526/27583309/stz039.pdf>. URL: <https://doi.org/10.1093/mnras/stz039>.
- [150] F Rastegarnia et al. “Deep learning in searching the spectroscopic redshift of quasars”. In: *Monthly Notices of the Royal Astronomical Society* 511.3 (Jan. 2022), pp. 4490–4499. ISSN: 0035-8711. DOI: 10.1093/mnras/stac076. eprint: <https://academic.oup.com/mnras/article-pdf/511/3/4490/42620742/stac076.pdf>. URL: <https://doi.org/10.1093/mnras/stac076>.
- [151] Nicholas Choma et al. “Graph Neural Networks for IceCube Signal Classification”. In: (2018), pp. 386–391. DOI: 10.1109/ICMLA.2018.00064.
- [152] R. Abbasi et al. “Graph Neural Networks for low-energy event classification & reconstruction in IceCube”. In: *Journal of Instrumentation* 17.11 (2022), P11003. DOI: 10.1088/1748-0221/17/11/P11003. URL: <https://dx.doi.org/10.1088/1748-0221/17/11/P11003>.
- [153] Tim Ruhe. “Application of machine learning algorithms in imaging Cherenkov and neutrino astronomy”. In: *International Journal of Modern Physics A* 35.33 (2020), p. 2043004. DOI: 10.1142/S0217751X20430046. eprint: <https://doi.org/10.1142/S0217751X20430046>. URL: <https://doi.org/10.1142/S0217751X20430046>.
- [154] Mohammad Kordzanganeh et al. “Quantum Machine Learning for Radio Astronomy”. In: *35th Conference on Neural Information Processing Systems*. Dec. 2021. arXiv: 2112.02655 [quant-ph].
- [155] Michelle Ntampaka et al. “Building Trustworthy Machine Learning Models for Astronomy”. In: *31st annual conference on Astronomical Data Analysis Software and Systems*. Nov. 2021. arXiv: 2111.14566 [astro-ph.IM].
- [156] Juan de Dios Rojas Olvera et al. “Observational Cosmology with Artificial Neural Networks”. In: *Universe* 8.2 (2022). ISSN: 2218-1997. DOI: 10.3390/universe8020120. URL: <https://www.mdpi.com/2218-1997/8/2/120>.

- [157] Kana Moriwaki et al. “Machine learning for observational cosmology”. In: *Reports on Progress in Physics* 86.7 (2023), p. 076901. DOI: 10.1088/1361-6633/acd2ea. URL: <https://dx.doi.org/10.1088/1361-6633/acd2ea>.
- [158] Cora Dvorkin et al. “Machine Learning and Cosmology”. In: *Snowmass 2021*. Mar. 2022. arXiv: 2203.08056 [hep-ph].
- [159] Dimitris K Iakovidis et al. “Roadmap on signal processing for next generation measurement systems”. In: *Measurement Science and Technology* 33.1 (2021), p. 012002. DOI: 10.1088/1361-6501/ac2dbd. URL: <https://dx.doi.org/10.1088/1361-6501/ac2dbd>.
- [160] Serkan Kiranyaz et al. “1-D Convolutional Neural Networks for Signal Processing Applications”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 8360–8364. DOI: 10.1109/ICASSP.2019.8682194.
- [161] Tara N. Sainath et al. “Multichannel Signal Processing With Deep Neural Networks for Automatic Speech Recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.5 (2017), pp. 965–979. DOI: 10.1109/TASLP.2017.2672401.
- [162] Daniel George et al. “Deep neural networks to enable real-time multimessenger astrophysics”. In: *Phys. Rev. D* 97 (4 2018), p. 044039. DOI: 10.1103/PhysRevD.97.044039. URL: <https://link.aps.org/doi/10.1103/PhysRevD.97.044039>.
- [163] Hunter Gabbard et al. “Matching Matched Filtering with Deep Networks for Gravitational-Wave Astronomy”. In: *Phys. Rev. Lett.* 120 (14 2018), p. 141103. DOI: 10.1103/PhysRevLett.120.141103. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.120.141103>.
- [164] Timothy D. Gebhard et al. “Convolutional neural networks: A magic bullet for gravitational-wave detection?” In: *Phys. Rev. D* 100 (6 2019), p. 063015. DOI: 10.1103/PhysRevD.100.063015. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.063015>.
- [165] Xiang-Ru Li et al. “Some optimizations on detecting gravitational wave using convolutional neural network”. In: *Frontiers of Physics* 15.5 (June 2020). DOI: 10.1007/s11467-020-0966-4. URL: <https://doi.org/10.1007/s11467-020-0966-4>.

- [166] Plamen G. Krastev. “Real-time detection of gravitational waves from binary neutron stars using artificial neural networks”. In: *Physics Letters B* 803 (2020), p. 135330. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2020.135330>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269320301349>.
- [167] Marlin B. Schäfer et al. “Detection of gravitational-wave signals from binary neutron star mergers using machine learning”. In: *Phys. Rev. D* 102 (6 2020), p. 063015. DOI: 10.1103/PhysRevD.102.063015. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.063015>.
- [168] Grégory Baltus et al. “Convolutional neural networks for the detection of the early inspiral of a gravitational-wave signal”. In: *Phys. Rev. D* 103 (10 2021), p. 102003. DOI: 10.1103/PhysRevD.103.102003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.102003>.
- [169] Wathela Alhassan et al. “Detection of Einstein telescope gravitational wave signals from binary black holes using deep learning”. In: *Monthly Notices of the Royal Astronomical Society* 519.3 (Dec. 2022), pp. 3843–3850. ISSN: 0035-8711. DOI: 10.1093/mnras/stac3797. eprint: <https://academic.oup.com/mnras/article-pdf/519/3/3843/48617338/stac3797.pdf>. URL: <https://doi.org/10.1093/mnras/stac3797>.
- [170] Roberto Corizzo et al. “Scalable auto-encoders for gravitational waves detection from time series data”. In: *Expert Systems with Applications* 151 (2020), p. 113378. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2020.113378>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417420301986>.
- [171] Yu-Chiung Lin et al. “Detection of gravitational waves using Bayesian neural networks”. In: *Phys. Rev. D* 103 (6 2021), p. 063034. DOI: 10.1103/PhysRevD.103.063034. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.063034>.
- [172] Wei Wei et al. “Deep learning for gravitational wave forecasting of neutron star mergers”. In: *Physics Letters B* 816 (2021), p. 136185. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2021.136185>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269321001258>.

-
- [173] Wei Wei et al. “Deep learning ensemble for real-time gravitational wave detection of spinning binary black hole mergers”. In: *Physics Letters B* 812 (2021), p. 136029. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2020.136029>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269320308327>.
- [174] Vasileios Skliris. “Machine Learning To Extract Gravitational Wave Transients”. PhD thesis. Cardiff U., 2021.
- [175] Shang-Jie Jin et al. “Rapid identification of time-frequency domain gravitational wave signals from binary black holes using deep learning”. In: (May 2023). arXiv: 2305.19003 [gr-qc].
- [176] Chayan Chatterjee et al. “Extraction of binary black hole gravitational wave signals from detector data using deep learning”. In: *Phys. Rev. D* 104 (6 2021), p. 064046. DOI: 10.1103/PhysRevD.104.064046. URL: <https://link.aps.org/doi/10.1103/PhysRevD.104.064046>.
- [177] Christoph Dreissigacker et al. “Deep-learning continuous gravitational waves: Multiple detectors and realistic noise”. In: *Phys. Rev. D* 102 (2 2020), p. 022005. DOI: 10.1103/PhysRevD.102.022005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.022005>.
- [178] Joe Bayley et al. “Robust machine learning algorithm to search for continuous gravitational waves”. In: *Phys. Rev. D* 102 (8 2020), p. 083024. DOI: 10.1103/PhysRevD.102.083024. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.083024>.
- [179] L. Sun et al. “Hidden Markov model tracking of continuous gravitational waves from young supernova remnants”. In: *Phys. Rev. D* 97 (4 2018), p. 043013. DOI: 10.1103/PhysRevD.97.043013. URL: <https://link.aps.org/doi/10.1103/PhysRevD.97.043013>.
- [180] Andrei Utina et al. “Deep learning searches for gravitational wave stochastic backgrounds”. In: *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*. 2021, pp. 1–6. DOI: 10.1109/CBMI50038.2021.9461904.
- [181] Takahiro S. Yamamoto et al. “Deep learning for intermittent gravitational wave signals”. In: *Phys. Rev. D* 107 (4 2023), p. 044032. DOI: 10.1103/PhysRevD.107.044032. URL: <https://link.aps.org/doi/10.1103/PhysRevD.107.044032>.

- [182] Alberto Iess et al. “Core-Collapse supernova gravitational-wave search and deep learning classification”. In: *Machine Learning: Science and Technology* 1.2 (2020), p. 025014. DOI: 10.1088/2632-2153/ab7d31. URL: <https://dx.doi.org/10.1088/2632-2153/ab7d31>.
- [183] Iess, Alberto et al. “LSTM and CNN application for core-collapse supernova search in gravitational wave real data”. In: *A&A* 669 (2023), A42. DOI: 10.1051/0004-6361/202142525. URL: <https://doi.org/10.1051/0004-6361/202142525>.
- [184] Marek J. Szczepańczyk et al. “Search for gravitational-wave bursts in the third Advanced LIGO-Virgo run with coherent WaveBurst enhanced by machine learning”. In: *Phys. Rev. D* 107 (6 2023), p. 062002. DOI: 10.1103/PhysRevD.107.062002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.107.062002>.
- [185] Luana M. Modafferi et al. “Convolutional neural network search for long-duration transient gravitational waves from glitching pulsars”. In: (Mar. 2023). arXiv: 2303.16720 [astro-ph.HE].
- [186] Dixeena Lopez et al. “Utilizing Gaussian mixture models in all-sky searches for short-duration gravitational wave bursts”. In: *Phys. Rev. D* 105 (6 2022), p. 063024. DOI: 10.1103/PhysRevD.105.063024. URL: <https://link.aps.org/doi/10.1103/PhysRevD.105.063024>.
- [187] Eric A Moreno et al. “Source-agnostic gravitational-wave detection with recurrent autoencoders”. In: *Machine Learning: Science and Technology* 3.2 (2022), p. 025001. DOI: 10.1088/2632-2153/ac5435. URL: <https://dx.doi.org/10.1088/2632-2153/ac5435>.
- [188] Tom Marianer et al. “A semisupervised machine learning search for never-seen gravitational-wave sources”. In: *Monthly Notices of the Royal Astronomical Society* 500.4 (Nov. 2020), pp. 5408–5419. ISSN: 0035-8711. DOI: 10.1093/mnras/staa3550. eprint: <https://academic.oup.com/mnras/article-pdf/500/4/5408/34925881/staa3550.pdf>. URL: <https://doi.org/10.1093/mnras/staa3550>.
- [189] Alvin J. K. Chua et al. “Learning Bayesian posteriors with neural networks for gravitational-wave inference”. In: *Phys. Rev. Lett.* 124.4 (2020), p. 041102. DOI: 10.1103/PhysRevLett.124.041102. arXiv: 1909.05966 [gr-qc].

- [190] Stephen R. Green et al. “Gravitational-wave parameter estimation with autoregressive neural network flows”. In: *Phys. Rev. D* 102 (10 2020), p. 104057. DOI: 10.1103/PhysRevD.102.104057. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.104057>.
- [191] Hongyu Shen et al. “Statistically-informed deep learning for gravitational wave parameter estimation”. In: *Machine Learning: Science and Technology* 3.1 (2021), p. 015007. DOI: 10.1088/2632-2153/ac3843. URL: <https://dx.doi.org/10.1088/2632-2153/ac3843>.
- [192] Plamen G. Krastev et al. “Detection and parameter estimation of gravitational waves from binary neutron-star mergers in real LIGO data using deep learning”. In: *Physics Letters B* 815 (2021), p. 136161. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2021.136161>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269321001015>.
- [193] João D Álvares et al. “Exploring gravitational-wave detection and parameter inference using deep learning methods”. In: *Classical and Quantum Gravity* 38.15 (2021), p. 155010. DOI: 10.1088/1361-6382/ac0455. URL: <https://dx.doi.org/10.1088/1361-6382/ac0455>.
- [194] Philippe Bacon et al. “Denoising gravitational-wave signals from binary black holes with dilated convolutional autoencoder”. In: *Machine Learning: Science and Technology* (2023). URL: <http://iopscience.iop.org/article/10.1088/2632-2153/acd90f>.
- [195] Hongyu 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)*. IEEE, May 2019. DOI: 10.1109/icassp.2019.8683061. URL: <https://doi.org/10.1109/icassp.2019.8683061>.
- [196] Wei Wei et al. “Gravitational wave denoising of binary black hole mergers with deep learning”. In: *Physics Letters B* 800 (2020), p. 135081. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2019.135081>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269319308032>.
- [197] Alvin J. K. Chua et al. “Reduced-Order Modeling with Artificial Neurons for Gravitational-Wave Inference”. In: *Phys. Rev. Lett.* 122 (21 2019), p. 211101. DOI: 10.1103/PhysRevLett.122.211101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.122.211101>.

- [198] Stefano Schmidt et al. “Machine learning gravitational waves from binary black hole mergers”. In: *Phys. Rev. D* 103 (4 2021), p. 043020. DOI: 10.1103/PhysRevD.103.043020. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.043020>.
- [199] Sebastian Khan et al. “Gravitational-wave surrogate models powered by artificial neural networks”. In: *Phys. Rev. D* 103 (6 2021), p. 064015. DOI: 10.1103/PhysRevD.103.064015. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.064015>.
- [200] Paraskevi Nousi et al. “Autoencoder-driven spiral representation learning for gravitational wave surrogate modelling”. In: *Neurocomputing* 491 (2022), pp. 67–77. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2022.03.052>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231222003472>.
- [201] Matteo Breschi et al. “Bayesian inference of multimessenger astrophysical data: Methods and applications to gravitational waves”. In: *Phys. Rev. D* 104 (4 2021), p. 042001. DOI: 10.1103/PhysRevD.104.042001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.104.042001>.
- [202] Soichiro Morisaki et al. “Rapid parameter estimation of gravitational waves from binary neutron star coalescence using focused reduced order quadrature”. In: *Phys. Rev. D* 102 (10 2020), p. 104020. DOI: 10.1103/PhysRevD.102.104020. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.104020>.
- [203] Dario Jozinović et al. “Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network”. In: *Geophysical Journal International* 222.2 (May 2020), pp. 1379–1389. ISSN: 0956-540X. DOI: 10.1093/gji/ggaa233. eprint: <https://academic.oup.com/gji/article-pdf/222/2/1379/33381513/ggaa233.pdf>. URL: <https://doi.org/10.1093/gji/ggaa233>.
- [204] Dario Jozinović et al. “Transfer learning: improving neural network based prediction of earthquake ground shaking for an area with insufficient training data”. In: *Geophysical Journal International* 229.1 (Dec. 2021), pp. 704–718. ISSN: 0956-540X. DOI: 10.1093/gji/ggab488. eprint: <https://academic.oup.com/gji/article-pdf/229/1/704/42238366/ggab488.pdf>. URL: <https://doi.org/10.1093/gji/ggab488>.

- [205] F Badaracco et al. “Machine learning for gravitational-wave detection: surrogate Wiener filtering for the prediction and optimized cancellation of Newtonian noise at Virgo”. In: *Classical and Quantum Gravity* 37.19 (2020), p. 195016. DOI: 10.1088/1361-6382/abab64. URL: <https://dx.doi.org/10.1088/1361-6382/abab64>.
- [206] A. Michelini et al. “INSTANCE – the Italian seismic dataset for machine learning”. In: *Earth System Science Data* 13.12 (2021), pp. 5509–5544. DOI: 10.5194/essd-13-5509-2021. URL: <https://essd.copernicus.org/articles/13/5509/2021/>.
- [207] Nikhil Mukund et al. “First demonstration of neural sensing and control in a kilometer-scale gravitational wave observatory”. In: (Jan. 2023). arXiv: 2301.06221 [physics.ins-det].
- [208] Matthew Mould et al. “Deep learning and Bayesian inference of gravitational-wave populations: Hierarchical black-hole mergers”. In: *Phys. Rev. D* 106 (10 2022), p. 103013. DOI: 10.1103/PhysRevD.106.103013. URL: <https://link.aps.org/doi/10.1103/PhysRevD.106.103013>.
- [209] E. A. Huerta et al. “Advances in Machine and Deep Learning for Modeling and Real-Time Detection of Multi-messenger Sources”. In: *Handbook of Gravitational Wave Astronomy*. Ed. by Cosimo Bambi et al. Singapore: Springer Singapore, 2020, pp. 1–27. ISBN: 978-981-15-4702-7. DOI: 10.1007/978-981-15-4702-7_47-1. URL: https://doi.org/10.1007/978-981-15-4702-7_47-1.
- [210] Elena Cuoco et al. “Enhancing gravitational-wave science with machine learning”. In: *Machine Learning: Science and Technology* 2.1 (2020), p. 011002. DOI: 10.1088/2632-2153/abb93a. URL: <https://dx.doi.org/10.1088/2632-2153/abb93a>.
- [211] G2net.eu. *G2net*. 2018. URL: <https://www.g2net.eu/> (visited on 01/16/2023).
- [212] Chris Messenger et al. *G2Net Gravitational Wave Detection*. 2021. URL: <https://kaggle.com/competitions/g2net-gravitational-wave-detection>.
- [213] Chris Messenger et al. *G2Net Detecting Continuous Gravitational Waves*. 2022. URL: <https://kaggle.com/competitions/g2net-detecting-continuous-gravitational-waves>.

- [214] Rodrigo Tenorio et al. *Learning to detect continuous gravitational waves*. Tech. rep. LIGO-P2200295. 2022. URL: <https://dcc.ligo.org/P2200295>.
- [215] Marlin B. Schäfer et al. “Training strategies for deep learning gravitational-wave searches”. In: *Phys. Rev. D* 105 (4 2022), p. 043002. DOI: 10.1103/PhysRevD.105.043002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.105.043002>.
- [216] Marlin B. Schäfer et al. “First machine learning gravitational-wave search mock data challenge”. In: *Phys. Rev. D* 107 (2 2023), p. 023021. DOI: 10.1103/PhysRevD.107.023021. URL: <https://link.aps.org/doi/10.1103/PhysRevD.107.023021>.
- [217] Albert Einstein. “Zur elektrodynamik bewegter körper”. In: *Annalen der physik* 4 (1905).
- [218] A. Einstein. “Die Grundlage der allgemeinen Relativitätstheorie”. In: *Annalen der Physik* 354.7 (1916), pp. 769–822. DOI: <https://doi.org/10.1002/andp.19163540702>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.19163540702>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19163540702>.
- [219] F. W. Dyson et al. “A Determination of the Deflection of Light by the Sun’s Gravitational Field, from Observations Made at the Total Eclipse of May 29, 1919”. In: *Phil. Trans. Roy. Soc. Lond. A* 220 (1920), pp. 291–333. DOI: 10.1098/rsta.1920.0009.
- [220] G. M. Clemence. “The Relativity Effect in Planetary Motions”. In: *Rev. Mod. Phys.* 19 (4 1947), pp. 361–364. DOI: 10.1103/RevModPhys.19.361. URL: <https://link.aps.org/doi/10.1103/RevModPhys.19.361>.
- [221] J. L. Greenstein et al. “Effective Temperature, Radius, and Gravitational Redshift of Sirius B”. In: *The Astrophysical Journal* 169 (Nov. 1971), p. 563. DOI: 10.1086/151174.
- [222] Charles W. Misner et al. *Gravitation*. San Francisco: W. H. Freeman, 1973. ISBN: 978-0-7167-0344-0, 978-0-691-17779-3.
- [223] Francesco Lacava. “Energy and Momentum of the Electromagnetic Field”. In: *Classical Electrodynamics: From Image Charges to the Photon Mass and Magnetic Monopoles*. Cham: Springer International Publishing, 2016, pp. 121–142.

- ISBN: 978-3-319-39474-9. DOI: 10.1007/978-3-319-39474-9_9. URL: https://doi.org/10.1007/978-3-319-39474-9_9.
- [224] Junjie Zhao et al. “Probing dipole radiation from binary neutron stars with ground-based laser-interferometer and atom-interferometer gravitational-wave observatories”. In: *Phys. Rev. D* 104 (8 2021), p. 084008. DOI: 10.1103/PhysRevD.104.084008. URL: <https://link.aps.org/doi/10.1103/PhysRevD.104.084008>.
- [225] Ziming Wang et al. “Simultaneous bounds on the gravitational dipole radiation and varying gravitational constant from compact binary inspirals”. In: *Physics Letters B* 834 (2022), p. 137416. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2022.137416>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269322005500>.
- [226] Karl Schwarzschild. “Über das Gravitationsfeld eines Massenpunktes nach der Einsteinschen Theorie”. In: *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften* (Jan. 1916), pp. 189–196.
- [227] Chris L Fryer. “Compact object formation and the supernova explosion engine”. In: *Classical and Quantum Gravity* 30.24 (2013), p. 244002. DOI: 10.1088/0264-9381/30/24/244002. URL: <https://dx.doi.org/10.1088/0264-9381/30/24/244002>.
- [228] A. Heger et al. “How Massive Single Stars End Their Life”. In: *The Astrophysical Journal* 591.1 (2003), p. 288. DOI: 10.1086/375341. URL: <https://dx.doi.org/10.1086/375341>.
- [229] J. R. Oppenheimer et al. “On Massive Neutron Cores”. In: *Phys. Rev.* 55 (4 1939), pp. 374–381. DOI: 10.1103/PhysRev.55.374. URL: <https://link.aps.org/doi/10.1103/PhysRev.55.374>.
- [230] Luciano Rezzolla et al. “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. DOI: 10.3847/2041-8213/aaa401. URL: <https://dx.doi.org/10.3847/2041-8213/aaa401>.
- [231] Maselli, A. et al. “Binary white dwarfs and decihertz gravitational wave observations: From the Hubble constant to supernova astrophysics”. In: *A&A* 635 (2020), A120. DOI: 10.1051/0004-6361/201936848. URL: <https://doi.org/10.1051/0004-6361/201936848>.

- [232] Francesca Matteucci. “Modelling the chemical evolution of the Milky Way”. In: *The Astronomy and Astrophysics Review* 29.1 (Aug. 2021). DOI: 10.1007/s00159-021-00133-8. URL: <https://doi.org/10.1007/s00159-021-00133-8>.
- [233] F.-K. Thielemann et al. “Neutron Star Mergers and Nucleosynthesis of Heavy Elements”. In: *Annual Review of Nuclear and Particle Science* 67.1 (2017), pp. 253–274. DOI: 10.1146/annurev-nucl-101916-123246. eprint: <https://doi.org/10.1146/annurev-nucl-101916-123246>. URL: <https://doi.org/10.1146/annurev-nucl-101916-123246>.
- [234] Roy P. Kerr. “Gravitational Field of a Spinning Mass as an Example of Algebraically Special Metrics”. In: *Phys. Rev. Lett.* 11 (5 1963), pp. 237–238. DOI: 10.1103/PhysRevLett.11.237. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.11.237>.
- [235] Bernd Bruegmann. “Binary black hole mergers in 3-d numerical relativity”. In: *Int. J. Mod. Phys. D* 8 (1999), p. 85. DOI: 10.1142/S0218271899000080. arXiv: [gr-qc/9708035](https://arxiv.org/abs/gr-qc/9708035).
- [236] Frans Pretorius. “Evolution of Binary Black-Hole Spacetimes”. In: *Phys. Rev. Lett.* 95 (12 2005), p. 121101. DOI: 10.1103/PhysRevLett.95.121101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.95.121101>.
- [237] M. Campanelli et al. “Accurate Evolutions of Orbiting Black-Hole Binaries without Excision”. In: *Phys. Rev. Lett.* 96 (11 2006), p. 111101. DOI: 10.1103/PhysRevLett.96.111101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.96.111101>.
- [238] John G. Baker et al. “Gravitational-Wave Extraction from an Inspiring Configuration of Merging Black Holes”. In: *Phys. Rev. Lett.* 96 (11 2006), p. 111102. DOI: 10.1103/PhysRevLett.96.111102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.96.111102>.
- [239] U Sperhake et al. “11-orbit inspiral of a mass ratio 4:1 black-hole binary”. In: *Classical and Quantum Gravity* 28.13 (2011), p. 134004. DOI: 10.1088/0264-9381/28/13/134004. URL: <https://dx.doi.org/10.1088/0264-9381/28/13/134004>.
- [240] David Hilditch et al. “Compact binary evolutions with the Z4c formulation”. In: *Phys. Rev. D* 88 (8 2013), p. 084057. DOI: 10.1103/PhysRevD.88.084057. URL: <https://link.aps.org/doi/10.1103/PhysRevD.88.084057>.

- [241] Lawrence E. Kidder et al. “Black hole evolution by spectral methods”. In: *Phys. Rev. D* 62 (8 2000), p. 084032. DOI: 10.1103/PhysRevD.62.084032. URL: <https://link.aps.org/doi/10.1103/PhysRevD.62.084032>.
- [242] Roland Haas et al. *The Einstein Toolkit*. Version The “Sophie Kowalevski” release, ET_2022_11. To find out more, visit <http://einstein toolkit.org>. Nov. 2022. DOI: 10.5281/zenodo.7245853. URL: <https://doi.org/10.5281/zenodo.7245853>.
- [243] Thibault Damour et al. “Hamiltonian of two spinning compact bodies with next-to-leading order gravitational spin-orbit coupling”. In: *Phys. Rev. D* 77 (6 2008), p. 064032. DOI: 10.1103/PhysRevD.77.064032. URL: <https://link.aps.org/doi/10.1103/PhysRevD.77.064032>.
- [244] Jan Steinhoff et al. “Next-to-leading order gravitational spin(1)-spin(2) dynamics in Hamiltonian form”. In: *Phys. Rev. D* 77 (8 2008), p. 081501. DOI: 10.1103/PhysRevD.77.081501. URL: <https://link.aps.org/doi/10.1103/PhysRevD.77.081501>.
- [245] Alessandra Buonanno et al. “Transition from inspiral to plunge in precessing binaries of spinning black holes”. In: *Phys. Rev. D* 74 (10 2006), p. 104005. DOI: 10.1103/PhysRevD.74.104005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.74.104005>.
- [246] Misao Sasaki et al. “Analytic Black Hole Perturbation Approach to Gravitational Radiation”. In: *Living Reviews in Relativity* 6.1 (Nov. 2003). DOI: 10.12942/lrr-2003-6. URL: <https://doi.org/10.12942/lrr-2003-6>.
- [247] Luc Blanchet. “Gravitational Radiation from Post-Newtonian Sources and Inspiralling Compact Binaries”. In: *Living Reviews in Relativity* 17.1 (Feb. 2014). DOI: 10.12942/lrr-2014-2. URL: <https://doi.org/10.12942/lrr-2014-2>.
- [248] A. Buonanno et al. “Effective one-body approach to general relativistic two-body dynamics”. In: *Phys. Rev. D* 59 (8 1999), p. 084006. DOI: 10.1103/PhysRevD.59.084006. URL: <https://link.aps.org/doi/10.1103/PhysRevD.59.084006>.
- [249] Alessandra Buonanno et al. “Transition from inspiral to plunge in binary black hole coalescences”. In: *Phys. Rev. D* 62 (6 2000), p. 064015. DOI: 10.1103/PhysRevD.62.064015. URL: <https://link.aps.org/doi/10.1103/PhysRevD.62.064015>.

- [250] Alessandro Nagar et al. “TEOBResumS: Analytic systematics in next-generation of effective-one-body gravitational waveform models for future observations”. In: (Apr. 2023). arXiv: 2304.09662 [gr-qc].
- [251] Alessandro Nagar et al. “Time-domain effective-one-body gravitational waveforms for coalescing compact binaries with nonprecessing spins, tides, and self-spin effects”. In: *Phys. Rev. D* 98 (10 2018), p. 104052. DOI: 10.1103/PhysRevD.98.104052. URL: <https://link.aps.org/doi/10.1103/PhysRevD.98.104052>.
- [252] Alejandro Bohé et al. “Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors”. In: *Phys. Rev. D* 95 (4 2017), p. 044028. DOI: 10.1103/PhysRevD.95.044028. URL: <https://link.aps.org/doi/10.1103/PhysRevD.95.044028>.
- [253] Rossella Gamba et al. “Effective-one-body waveforms for precessing coalescing compact binaries with post-Newtonian twist”. In: *Phys. Rev. D* 106 (2 2022), p. 024020. DOI: 10.1103/PhysRevD.106.024020. URL: <https://link.aps.org/doi/10.1103/PhysRevD.106.024020>.
- [254] Antoni Ramos-Buades et al. “SEOBNRv5PHM: Next generation of accurate and efficient multipolar precessing-spin effective-one-body waveforms for binary black holes”. In: (Mar. 2023). arXiv: 2303.18046 [gr-qc].
- [255] P Ajith et al. “A phenomenological template family for black-hole coalescence waveforms”. In: *Classical and Quantum Gravity* 24.19 (2007), S689. DOI: 10.1088/0264-9381/24/19/S31. URL: <https://dx.doi.org/10.1088/0264-9381/24/19/S31>.
- [256] P. Ajith et al. “Template bank for gravitational waveforms from coalescing binary black holes: Nonspinning binaries”. In: *Phys. Rev. D* 77 (10 2008), p. 104017. DOI: 10.1103/PhysRevD.77.104017. URL: <https://link.aps.org/doi/10.1103/PhysRevD.77.104017>.
- [257] L. Santamaría et al. “Matching post-Newtonian and numerical relativity waveforms: Systematic errors and a new phenomenological model for nonprecessing black hole binaries”. In: *Phys. Rev. D* 82 (6 2010), p. 064016. DOI: 10.1103/PhysRevD.82.064016. URL: <https://link.aps.org/doi/10.1103/PhysRevD.82.064016>.

- [258] P. Ajith et al. “Inspirals-Merger-Ringdown Waveforms for Black-Hole Binaries with Nonprecessing Spins”. In: *Phys. Rev. Lett.* 106 (24 2011), p. 241101. DOI: 10.1103/PhysRevLett.106.241101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.106.241101>.
- [259] Sebastian Khan et al. “Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era”. In: *Phys. Rev. D* 93 (4 2016), p. 044007. DOI: 10.1103/PhysRevD.93.044007. URL: <https://link.aps.org/doi/10.1103/PhysRevD.93.044007>.
- [260] Geraint Pratten et al. “Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes”. In: *Phys. Rev. D* 103 (10 2021), p. 104056. DOI: 10.1103/PhysRevD.103.104056. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.104056>.
- [261] Scott E. Field et al. “Fast Prediction and Evaluation of Gravitational Waveforms Using Surrogate Models”. In: *Phys. Rev. X* 4 (3 2014), p. 031006. DOI: 10.1103/PhysRevX.4.031006. URL: <https://link.aps.org/doi/10.1103/PhysRevX.4.031006>.
- [262] Michael Pürrer. “Frequency-domain reduced order models for gravitational waves from aligned-spin compact binaries”. In: *Classical and Quantum Gravity* 31.19 (2014), p. 195010. DOI: 10.1088/0264-9381/31/19/195010. URL: <https://dx.doi.org/10.1088/0264-9381/31/19/195010>.
- [263] Jonathan Blackman et al. “Fast and Accurate Prediction of Numerical Relativity Waveforms from Binary Black Hole Coalescences Using Surrogate Models”. In: *Phys. Rev. Lett.* 115 (12 2015), p. 121102. DOI: 10.1103/PhysRevLett.115.121102. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.115.121102>.
- [264] Michael Pürrer. “Frequency domain reduced order model of aligned-spin effective-one-body waveforms with generic mass ratios and spins”. In: *Phys. Rev. D* 93 (6 2016), p. 064041. DOI: 10.1103/PhysRevD.93.064041. URL: <https://link.aps.org/doi/10.1103/PhysRevD.93.064041>.
- [265] Jonathan Blackman et al. “A Surrogate model of gravitational waveforms from numerical relativity simulations of precessing binary black hole mergers”. In: *Phys. Rev. D* 95 (10 2017), p. 104023. DOI: 10.1103/PhysRevD.95.104023. URL: <https://link.aps.org/doi/10.1103/PhysRevD.95.104023>.

- [266] Jonathan Blackman et al. “Numerical relativity waveform surrogate model for generically precessing binary black hole mergers”. In: *Phys. Rev. D* 96 (2 2017), p. 024058. DOI: 10.1103/PhysRevD.96.024058. URL: <https://link.aps.org/doi/10.1103/PhysRevD.96.024058>.
- [267] Vijay Varma et al. “Surrogate models for precessing binary black hole simulations with unequal masses”. In: *Phys. Rev. Res.* 1 (3 2019), p. 033015. DOI: 10.1103/PhysRevResearch.1.033015. URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.1.033015>.
- [268] Qianyun Yun et al. “Surrogate model for gravitational waveforms of spin-aligned binary black holes with eccentricities”. In: *Phys. Rev. D* 103 (12 2021), p. 124053. DOI: 10.1103/PhysRevD.103.124053. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.124053>.
- [269] Manuel Tiglio et al. “Reduced order and surrogate models for gravitational waves”. In: *Living Reviews in Relativity* 25.1 (Apr. 2022). DOI: 10.1007/s41114-022-00035-w. URL: <https://doi.org/10.1007/s41114-022-00035-w>.
- [270] Marissa Walker et al. “Numerical-relativity surrogate modeling with nearly extremal black-hole spins”. In: *Classical and Quantum Gravity* 40.5 (2023), p. 055003. DOI: 10.1088/1361-6382/acb3a7. URL: <https://dx.doi.org/10.1088/1361-6382/acb3a7>.
- [271] Michael Boyle et al. “The SXS collaboration catalog of binary black hole simulations”. In: *Classical and Quantum Gravity* 36.19 (2019), p. 195006. DOI: 10.1088/1361-6382/ab34e2. URL: <https://dx.doi.org/10.1088/1361-6382/ab34e2>.
- [272] Tim Dietrich et al. “Interpreting binary neutron star mergers: describing the binary neutron star dynamics, modelling gravitational waveforms, and analyzing detections”. In: *General Relativity and Gravitation* 53.3 (Mar. 2021). DOI: 10.1007/s10714-020-02751-6. URL: <https://doi.org/10.1007/s10714-020-02751-6>.
- [273] Alejandra Gonzalez et al. “Second release of the CoRe database of binary neutron star merger waveforms”. In: *Classical and Quantum Gravity* 40.8 (2023), p. 085011. DOI: 10.1088/1361-6382/acc231. URL: <https://dx.doi.org/10.1088/1361-6382/acc231>.

- [274] K. Riles. “Gravitational waves: Sources, detectors and searches”. In: *Progress in Particle and Nuclear Physics* 68 (2013), pp. 1–54. ISSN: 0146-6410. DOI: <https://doi.org/10.1016/j.pnnp.2012.08.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0146641012001093>.
- [275] Bernard F Schutz. “Networks of gravitational wave detectors and three figures of merit”. In: *Classical and Quantum Gravity* 28.12 (2011), p. 125023. DOI: 10.1088/0264-9381/28/12/125023. URL: <https://dx.doi.org/10.1088/0264-9381/28/12/125023>.
- [276] B. P. Abbott et al. “GW150914: The Advanced LIGO Detectors in the Era of First Discoveries”. In: *Phys. Rev. Lett.* 116.13 (2016), p. 131103. DOI: 10.1103/PhysRevLett.116.131103. arXiv: 1602.03838 [gr-qc].
- [277] Benjamin P. Abbott et al. “Sensitivity of the Advanced LIGO detectors at the beginning of gravitational wave astronomy”. In: *Phys. Rev. D* 93.11 (2016). [Addendum: *Phys.Rev.D* 97, 059901 (2018)], p. 112004. DOI: 10.1103/PhysRevD.93.112004. arXiv: 1604.00439 [astro-ph.IM].
- [278] The LIGO Scientific Collaboration et al. “Advanced LIGO”. In: *Classical and Quantum Gravity* 32.7 (2015), p. 074001. DOI: 10.1088/0264-9381/32/7/074001. URL: <https://dx.doi.org/10.1088/0264-9381/32/7/074001>.
- [279] S Chatterji et al. “Multiresolution techniques for the detection of gravitational-wave bursts”. In: *Classical and Quantum Gravity* 21.20 (2004), S1809. DOI: 10.1088/0264-9381/21/20/024. URL: <https://dx.doi.org/10.1088/0264-9381/21/20/024>.
- [280] Derek Davis et al. *LIGO Document T2100045-v2*. 2021. URL: <https://dcc.ligo.org/LIGO-T2100045/public> (visited on 05/23/2023).
- [281] Irene Fiori et al. “The Hunt for Environmental Noise in Virgo during the Third Observing Run”. In: *Galaxies* 8.4 (2020). ISSN: 2075-4434. DOI: 10.3390/galaxies8040082. URL: <https://www.mdpi.com/2075-4434/8/4/82>.
- [282] F. Acernese et al. “Virgo Detector Characterization and Data Quality: tools”. In: (Oct. 2022). arXiv: 2210.15634 [gr-qc].
- [283] F. Acernese et al. “Virgo Detector Characterization and Data Quality: results from the O3 run”. In: (Oct. 2022). arXiv: 2210.15633 [gr-qc].

- [284] Derek Davis et al. “LIGO detector characterization in the second and third observing runs”. In: *Classical and Quantum Gravity* 38.13 (2021), p. 135014. DOI: 10.1088/1361-6382/abfd85. URL: <https://dx.doi.org/10.1088/1361-6382/abfd85>.
- [285] Florent Robinet et al. “Omicron: a tool to characterize transient noise in gravitational-wave detectors”. In: *SoftwareX* 12 (2020), p. 100620. DOI: 10.1016/j.softx.2020.100620. arXiv: 2007.11374 [astro-ph.IM].
- [286] Alexander H Nitz. “Distinguishing short duration noise transients in LIGO data to improve the PyCBC search for gravitational waves from high mass binary black hole mergers”. In: *Classical and Quantum Gravity* 35.3 (2018), p. 035016. DOI: 10.1088/1361-6382/aaa13d. URL: <https://dx.doi.org/10.1088/1361-6382/aaa13d>.
- [287] B P Abbott et al. “Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO’s first observing run”. In: *Classical and Quantum Gravity* 35.6 (2018), p. 065010. DOI: 10.1088/1361-6382/aaaafa. URL: <https://dx.doi.org/10.1088/1361-6382/aaaafa>.
- [288] F Acernese et al. “The Virgo O3 run and the impact of the environment”. In: *Classical and Quantum Gravity* 39.23 (2022), p. 235009. DOI: 10.1088/1361-6382/ac776a. URL: <https://dx.doi.org/10.1088/1361-6382/ac776a>.
- [289] Michael Zevin et al. “Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science”. In: *Class. Quant. Grav.* 34.6 (2017), p. 064003. DOI: 10.1088/1361-6382/aa5cea. arXiv: 1611.04596 [gr-qc].
- [290] Daniel George et al. “Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO”. In: (June 2017). arXiv: 1706.07446 [gr-qc].
- [291] S. Bahaadini et al. “Machine learning for Gravity Spy: Glitch classification and dataset”. In: *Info. Sci.* 444 (2018), pp. 172–186. DOI: 10.1016/j.ins.2018.02.068.
- [292] J Glanzer et al. “Data quality up to the third observing run of advanced LIGO: Gravity Spy glitch classifications”. In: *Classical and Quantum Gravity* 40.6 (2023), p. 065004. DOI: 10.1088/1361-6382/acb633. URL: <https://dx.doi.org/10.1088/1361-6382/acb633>.

- [293] Robert E. Colgan et al. “Efficient gravitational-wave glitch identification from environmental data through machine learning”. In: *Phys. Rev. D* 101 (10 2020), p. 102003. DOI: 10.1103/PhysRevD.101.102003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.101.102003>.
- [294] Gravitational Wave Open Science Center. *Strain Data Release for GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs*. 2018. DOI: 10.7935/82H3-HH23. URL: <https://www.gw-openscience.org/GWTC-1/>.
- [295] Tiago S. Fernandes et al. “Convolutional Neural Networks for the classification of glitches in gravitational-wave data streams”. In: (Mar. 2023). arXiv: 2303.13917 [gr-qc].
- [296] “LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S”. In: *Philosophical Transactions of the Royal Society of London* 53 (Dec. 1763), pp. 370–418. DOI: 10.1098/rstl.1763.0053. URL: <https://doi.org/10.1098/rstl.1763.0053>.
- [297] David M. W. Powers. “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation”. In: (2020). DOI: 10.48550/ARXIV.2010.16061. URL: <https://arxiv.org/abs/2010.16061>.
- [298] Devidas V Pai et al. *Fundamentals of approximation theory*. Boca Raton, FL: CRC Press, Sept. 2000.
- [299] I.A Basheer et al. “Artificial neural networks: fundamentals, computing, design, and application”. In: *Journal of Microbiological Methods* 43.1 (2000). Neural Computing in Micrbiology, pp. 3–31. ISSN: 0167-7012. DOI: [https://doi.org/10.1016/S0167-7012\(00\)00201-3](https://doi.org/10.1016/S0167-7012(00)00201-3). URL: <https://www.sciencedirect.com/science/article/pii/S0167701200002013>.
- [300] Han-Sheng Wang et al. “A machine learning study to identify collective flow in small and large colliding systems”. In: (May 2023). arXiv: 2305.09937 [nucl-th].
- [301] Dominik Koutný et al. “Neural-network quantum state tomography”. In: *Phys. Rev. A* 106 (1 2022), p. 012409. DOI: 10.1103/PhysRevA.106.012409. URL: <https://link.aps.org/doi/10.1103/PhysRevA.106.012409>.

- [302] Chris Whittle et al. “Machine Learning for Quantum-Enhanced Gravitational-Wave Observatories”. In: (May 2023). arXiv: 2305.13780 [astro-ph.IM].
- [303] C. H. Kim et al. “Restoring Original Signal From Pile-up Signal using Deep Learning”. In: (Apr. 2023). arXiv: 2304.14496 [physics.ins-det].
- [304] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (Feb. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2. URL: <https://doi.org/10.1038/s41592-019-0686-2>.
- [305] Thomas Mitchell. *Machine Learning*. en. McGraw-Hill series in computer science. New York, NY: McGraw-Hill Professional, Mar. 1997.
- [306] Ken-Ichi Funahashi. “On the approximate realization of continuous mappings by neural networks”. In: *Neural Networks* 2.3 (1989), pp. 183–192. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(89\)90003-8](https://doi.org/10.1016/0893-6080(89)90003-8). URL: <https://www.sciencedirect.com/science/article/pii/0893608089900038>.
- [307] Vitaly Maiorov et al. “Lower bounds for approximation by MLP neural networks”. In: *Neurocomputing* 25.1 (1999), pp. 81–91. ISSN: 0925-2312. DOI: [https://doi.org/10.1016/S0925-2312\(98\)00111-8](https://doi.org/10.1016/S0925-2312(98)00111-8). URL: <https://www.sciencedirect.com/science/article/pii/S0925231298001118>.
- [308] D.H. Wolpert et al. “No free lunch theorems for optimization”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: 10.1109/4235.585893.
- [309] K. Kawaguchi et al. “Generalization in Deep Learning”. In: *Mathematical Aspects of Deep Learning*. Ed. by Philipp Grohs et al. Cambridge University Press, 2022, 112–148. DOI: 10.1017/9781009025096.003.
- [310] Tor Lattimore et al. “No Free Lunch versus Occam’s Razor in Supervised Learning”. In: *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence: Papers from the Ray Solomonoff 85th Memorial Conference, Melbourne, VIC, Australia, November 30 – December 2, 2011*. Ed. by David L. Dowe. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 223–235. ISBN: 978-3-642-44958-1. DOI: 10.1007/978-3-642-44958-1_17. URL: https://doi.org/10.1007/978-3-642-44958-1_17.
- [311] Y. LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

-
- [312] John Duchi et al. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159. URL: <http://jmlr.org/papers/v12/duchi11a.html>.
- [313] Tijmen Tieleman et al. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31.
- [314] Diederik P. Kingma et al. “Adam: A Method for Stochastic Optimization”. In: *arXiv e-prints*, arXiv:1412.6980 (Dec. 2014), arXiv:1412.6980. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs.LG].
- [315] F. Alvarez et al. “Steepest Descent with Curvature Dynamical System”. In: *Journal of Optimization Theory and Applications* 120.2 (Feb. 2004), pp. 247–273.
- [316] R. Mohammadi Asl et al. “Optimal fractional order PID for a robotic manipulator using colliding bodies design”. In: *Soft Computing* 22.14 (2017), pp. 4647–4659.
- [317] Xavier Glorot et al. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh et al. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256. URL: <https://proceedings.mlr.press/v9/glorot10a.html>.
- [318] Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. arXiv: 1502.01852 [cs.CV].
- [319] James Martens. “Deep Learning via Hessian-Free Optimization”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML’10. Haifa, Israel: Omnipress, 2010, pp. 735–742. ISBN: 9781605589077.
- [320] A Saxe et al. “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks”. In: *International Conference on Learning Representations* 2014, 2014.

- [321] L. Y. Pratt. “Discriminability-Based Transfer between Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Hanson et al. Vol. 5. Morgan-Kaufmann, 1992. URL: https://proceedings.neurips.cc/paper_files/paper/1992/file/67e103b0761e60683e83c559be18d40c-Paper.pdf.
- [322] Chuanqi Tan et al. “A Survey on Deep Transfer Learning”. In: *Artificial Neural Networks and Machine Learning – ICANN 2018*. Ed. by Věra Kůrková et al. Cham: Springer International Publishing, 2018, pp. 270–279. ISBN: 978-3-030-01424-7.
- [323] E. Meijering. “A chronology of interpolation: from ancient astronomy to modern signal and image processing”. In: *Proceedings of the IEEE* 90.3 (2002), pp. 319–342. DOI: 10.1109/5.993400.
- [324] Stuart Geman et al. “Neural Networks and the Bias/Variance Dilemma”. In: *Neural Computation* 4.1 (Jan. 1992), pp. 1–58. ISSN: 0899-7667. DOI: 10.1162/neco.1992.4.1.1. eprint: <https://direct.mit.edu/neco/article-pdf/4/1/1/812244/neco.1992.4.1.1.pdf>. URL: <https://doi.org/10.1162/neco.1992.4.1.1>.
- [325] Christopher M. Bishop. “Regularization and complexity control in feed-forward networks”. English. In: *Proceedings International Conference on Artificial Neural Networks ICANN’95*. International Conference on Artificial Neural Networks ICANN’95. EC2 et Cie, 1995, pp. 141–148. ISBN: 2-910085-19-8.
- [326] J. Sjöberg et al. “Overtraining, regularization and searching for a minimum, with application to neural networks”. In: *International Journal of Control* 62.6 (1995), pp. 1391–1407. DOI: 10.1080/00207179508921605. eprint: <https://doi.org/10.1080/00207179508921605>. URL: <https://doi.org/10.1080/00207179508921605>.
- [327] Dulari Bhatt et al. “CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope”. In: *Electronics* 10.20 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10202470. URL: <https://www.mdpi.com/2079-9292/10/20/2470>.
- [328] Kaipeng Zhang et al. “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks”. In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503. DOI: 10.1109/LSP.2016.2603342.

- [329] Xuanhan Wang et al. “Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition”. In: *IEEE Signal Processing Letters* 24.4 (2017), pp. 510–514. DOI: 10.1109/LSP.2016.2611485.
- [330] Shuiwang Ji et al. “3D Convolutional Neural Networks for Human Action Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 221–231. DOI: 10.1109/TPAMI.2012.59.
- [331] Sanghyeop Lee et al. “Car plate recognition based on CNN using embedded system with GPU”. In: *2017 10th International Conference on Human System Interactions (HSI)*. 2017, pp. 239–241. DOI: 10.1109/HSI.2017.8005037.
- [332] Vijay Badrinarayanan et al. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.
- [333] I Sonata et al. “Autonomous car using CNN deep learning algorithm”. In: *Journal of Physics: Conference Series* 1869.1 (2021), p. 012071. DOI: 10.1088/1742-6596/1869/1/012071. URL: <https://dx.doi.org/10.1088/1742-6596/1869/1/012071>.
- [334] Akhil Agnihotri et al. *A Convolutional Neural Network Approach Towards Self-Driving Cars*. 2019. arXiv: 1909.03854 [cs.R0].
- [335] Meixin Zhu et al. “Human-like autonomous car-following model with deep reinforcement learning”. In: *Transportation Research Part C: Emerging Technologies* 97 (2018), pp. 348–368. ISSN: 0968-090X. DOI: <https://doi.org/10.1016/j.trc.2018.10.024>. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X1830055X>.
- [336] Takafumi Okuyama et al. “Autonomous Driving System based on Deep Q Learnig”. In: *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*. 2018, pp. 201–205. DOI: 10.1109/ICoIAS.2018.8494053.
- [337] Yann LeCun et al. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: 10.1038/nature14539. URL: <https://doi.org/10.1038/nature14539>.
- [338] David E. Rumelhart et al. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0. URL: <https://doi.org/10.1038/323533a0>.

- [339] Jan Kukačka et al. *Regularization for Deep Learning: A Taxonomy*. 2017. arXiv: 1710.10686 [cs.LG].
- [340] Razvan Pascanu et al. *On the difficulty of training Recurrent Neural Networks*. 2013. arXiv: 1211.5063 [cs.LG].
- [341] Razvan Pascanu et al. *On the difficulty of training Recurrent Neural Networks*. 2013. arXiv: 1211.5063 [cs.LG].
- [342] Jingzhao Zhang et al. *Why gradient clipping accelerates training: A theoretical justification for adaptivity*. 2020. arXiv: 1905.11881 [math.OC].
- [343] Connor Shorten et al. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* 6.1 (July 2019). DOI: 10.1186/s40537-019-0197-0. URL: <https://doi.org/10.1186/s40537-019-0197-0>.
- [344] Ekin D. Cubuk et al. *AutoAugment: Learning Augmentation Policies from Data*. 2019. arXiv: 1805.09501 [cs.CV].
- [345] Geoffrey E. Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv e-prints*, arXiv:1207.0580 (July 2012), arXiv:1207.0580. DOI: 10.48550/arXiv.1207.0580. arXiv: 1207.0580 [cs.NE].
- [346] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [347] Jonathan Tompson et al. “Efficient Object Localization Using Convolutional Networks”. In: *arXiv e-prints*, arXiv:1411.4280 (Nov. 2014), arXiv:1411.4280. DOI: 10.48550/arXiv.1411.4280. arXiv: 1411.4280 [cs.CV].
- [348] Petru Soviany et al. “Curriculum Learning: A Survey”. In: *International Journal of Computer Vision* 130.6 (Apr. 2022), pp. 1526–1565. DOI: 10.1007/s11263-022-01611-x. URL: <https://doi.org/10.1007/s11263-022-01611-x>.
- [349] Jeffrey L. Elman. “Learning and development in neural networks: the importance of starting small”. In: *Cognition* 48.1 (1993), pp. 71–99. ISSN: 0010-0277. DOI: [https://doi.org/10.1016/0010-0277\(93\)90058-4](https://doi.org/10.1016/0010-0277(93)90058-4). URL: <https://www.sciencedirect.com/science/article/pii/0010027793900584>.

- [350] 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, 41–48. ISBN: 9781605585161. DOI: 10.1145/1553374.1553380. URL: <https://doi.org/10.1145/1553374.1553380>.
- [351] Marlin B. Schäfer et al. *Training Strategies for Deep Learning Gravitational-Wave Searches*. 2021. URL: <https://github.com/gwastro/ml-training-strategies> (visited on 06/01/2023).
- [352] Marlin B. Schäfer et al. “From one to many: A deep learning coincident gravitational-wave search”. In: *Phys. Rev. D* 105 (4 2022), p. 043003. DOI: 10.1103/PhysRevD.105.043003. URL: <https://link.aps.org/doi/10.1103/PhysRevD.105.043003>.
- [353] Caleb Devine et al. “Optimizing spinning time-domain gravitational waveforms for advanced LIGO data analysis”. In: *Classical and Quantum Gravity* 33.12 (2016), p. 125025. DOI: 10.1088/0264-9381/33/12/125025. URL: <https://dx.doi.org/10.1088/0264-9381/33/12/125025>.
- [354] Alex Nitz et al. *gwastro/pycbc: v2.1.2 release of PyCBC*. Version v2.1.2. May 2023. DOI: 10.5281/zenodo.7885796. URL: <https://doi.org/10.5281/zenodo.7885796>.
- [355] LIGO Scientific Collaboration. *LIGO Algorithm Library - LALSuite*. free software (GPL). 2018. DOI: 10.7935/GT1W-FZ16. URL: <https://git.ligo.org/lscsoft/lalsuite>.
- [356] Karl Wette. “SWIGLAL: Python and Octave interfaces to the LALSuite gravitational-wave data analysis libraries”. In: *SoftwareX* 12 (July 2020), p. 100634. DOI: 10.1016/j.softx.2020.100634. URL: <https://doi.org/10.1016/j.softx.2020.100634>.
- [357] B. P. Abbott et al. “Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA”. In: *Living Reviews in Relativity* 23.1 (Sept. 2020). DOI: 10.1007/s41114-020-00026-9. URL: <https://doi.org/10.1007/s41114-020-00026-9>.
- [358] Enwenode Onajite. “Chapter 8 - Understanding Sample Data”. In: *Seismic Data Analysis Techniques in Hydrocarbon Exploration*. Ed. by Enwenode Onajite. Oxford: Elsevier, 2014, pp. 105–115. ISBN: 978-0-12-420023-4. DOI: <https://doi.org/10.1016/B978-0-12-420023-4.ch08>.

- doi.org/10.1016/B978-0-12-420023-4.00008-3. URL: <https://www.sciencedirect.com/science/article/pii/B9780124200234000083>.
- [359] Peter Fritschel. *LIGO Document T070247-v1*. 2009. URL: <https://dcc.ligo.org/LIGO-T070247/public> (visited on 03/31/2023).
- [360] David Shoemaker. *LIGO Document T0900288-v3*. 2018. URL: <https://dcc.ligo.org/LIGO-T0900288/public> (visited on 03/31/2023).
- [361] Sergey Ioffe et al. “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, 448–456.
- [362] R. Abbott et al. “Open data from the third observing run of LIGO, Virgo, KAGRA and GEO”. In: (Feb. 2023). arXiv: 2302.03676 [gr-qc].
- [363] Marlin Schäfer et al. *MLGWSC-1 Github repository*. 2021. URL: <https://github.com/gwastro/ml-mock-data-challenge-1> (visited on 02/23/2022).
- [364] Ondřej Zelenka. *ml-gw-search Github repository release v1.1*. 2023. URL: <https://github.com/ondrzal/ml-gw-search/tree/v1.1> (visited on 06/08/2023).
- [365] The HDF Group. *Hierarchical Data Format, version 5*. <https://www.hdfgroup.org/HDF5/1997-2022>.
- [366] B. Abbott et al. “Search for gravitational waves from binary inspirals in S3 and S4 LIGO data”. In: *Phys. Rev. D* 77 (2008), p. 062002. DOI: 10.1103/PhysRevD.77.062002. arXiv: 0704.3368 [gr-qc].
- [367] LIGO Scientific Collaboration et al. *LIGO Virgo strain data from GWTC-2 Catalog*. 2020. DOI: 10.7935/99GF-AX93. URL: <https://www.gw-openscience.org/GWTC-2>.
- [368] Hunter Gabbard et al. “Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy”. In: *Nature Physics* 18.1 (Dec. 2021), pp. 112–117. DOI: 10.1038/s41567-021-01425-7. URL: <https://doi.org/10.1038/s41567-021-01425-7>.
- [369] Rich Abbott et al. “Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo”. In: *SoftwareX* 13 (2021), p. 100658. DOI: 10.1016/j.softx.2021.100658. arXiv: 1912.11716 [gr-qc].

- [370] Daniel George et al. “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. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2017.12.053>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269317310390>.
- [371] cWB Development Team. *coherent WaveBurst Homepage*. 2021. URL: <https://gwburst.gitlab.io/>.
- [372] Sergey Klimenko et al. *cWB pipeline library: 6.4.1*. Version cWB-6.4.1. Dec. 2021. DOI: 10.5281/zenodo.5798976. URL: <https://doi.org/10.5281/zenodo.5798976>.
- [373] T. Cokelaer. “Gravitational waves from inspiralling compact binaries: Hexagonal template placement and its efficiency in detecting physical signals”. In: *Phys. Rev. D* 76 (10 2007), p. 102004. DOI: 10.1103/PhysRevD.76.102004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.76.102004>.
- [374] Michele Maggiore. *Gravitational Waves. Vol. 1: Theory and Experiments*. Oxford Master Series in Physics. Oxford University Press, 2007. ISBN: 978-0-19-857074-5, 978-0-19-852074-0.
- [375] Paraskevi Nousi et al. “Deep Residual Networks for Gravitational Wave Detection”. In: (Nov. 2022). arXiv: 2211.01520 [gr-qc].
- [376] Nikolaos Passalis et al. “Deep Adaptive Input Normalization for Time Series Forecasting”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.9 (2020), pp. 3760–3765. DOI: 10.1109/TNNLS.2019.2944933.
- [377] Paraskevi Nousi et al. *AResGW: Augmentation and RESidual networks for Gravitational Wave detection*. 2022. URL: <https://github.com/vivinousi/gw-detection-deep-learning> (visited on 06/01/2023).
- [378] Ben Margalit et al. “Constraining the Maximum Mass of Neutron Stars from Multi-messenger Observations of GW170817”. In: *The Astrophysical Journal Letters* 850.2 (2017), p. L19. DOI: 10.3847/2041-8213/aa991c. URL: <https://dx.doi.org/10.3847/2041-8213/aa991c>.
- [379] Dong-Sheng Shao et al. “Maximum mass cutoff in the neutron star mass distribution and the prospect of forming supramassive objects in the double neutron star mergers”. In: *Phys. Rev. D* 102 (6 2020), p. 063006. DOI: 10.1103/

- PhysRevD.102.063006. URL: <https://link.aps.org/doi/10.1103/PhysRevD.102.063006>.
- [380] Yeunhwan Lim et al. “Radius and equation of state constraints from massive neutron stars and GW190814”. In: *Phys. Rev. C* 104 (3 2021), p. L032802. DOI: 10.1103/PhysRevC.104.L032802. URL: <https://link.aps.org/doi/10.1103/PhysRevC.104.L032802>.
- [381] G. Raaijmakers et al. “Constraints on the Dense Matter Equation of State and Neutron Star Properties from NICER’s Mass–Radius Estimate of PSR J0740+6620 and Multimessenger Observations”. In: *The Astrophysical Journal Letters* 918.2 (2021), p. L29. DOI: 10.3847/2041-8213/ac089a. URL: <https://dx.doi.org/10.3847/2041-8213/ac089a>.
- [382] Rory J E Smith et al. “Massively parallel Bayesian inference for transient gravitational-wave astronomy”. In: *Monthly Notices of the Royal Astronomical Society* 498.3 (Aug. 2020), pp. 4492–4502. ISSN: 0035-8711. DOI: 10.1093/mnras/staa2483. eprint: <https://academic.oup.com/mnras/article-pdf/498/3/4492/33798799/staa2483.pdf>. URL: <https://doi.org/10.1093/mnras/staa2483>.
- [383] LIGO Scientific Collaboration et al. *LIGO Virgo strain data from GWTC-3 Catalog*. 2021. DOI: 10.7935/B024-1886. URL: <https://www.gw-openscience.org/GWTC-3>.
- [384] I. W. Harry et al. “Targeted coherent search for gravitational waves from compact binary coalescences”. In: *Phys. Rev. D* 83 (8 2011), p. 084002. DOI: 10.1103/PhysRevD.83.084002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.83.084002>.
- [385] I W Harry et al. “A coherent triggered search for single-spin compact binary coalescences in gravitational wave data”. In: *Classical and Quantum Gravity* 28.13 (2011), p. 134008. DOI: 10.1088/0264-9381/28/13/134008. URL: <https://dx.doi.org/10.1088/0264-9381/28/13/134008>.
- [386] D. Macleod et al. “Fully-coherent all-sky search for gravitational-waves from compact binary coalescences”. In: *Phys. Rev. D* 93 (6 2016), p. 064004. DOI: 10.1103/PhysRevD.93.064004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.93.064004>.

- [387] Sepp Hochreiter et al. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [388] Yuewei Zhang et al. “Deep learning model based on a bidirectional gated recurrent unit for the detection of gravitational wave signals”. In: *Phys. Rev. D* 106 (12 2022), p. 122002. DOI: 10.1103/PhysRevD.106.122002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.106.122002>.
- [389] Joongoo Lee et al. “Deep learning model on gravitational waveforms in merging and ringdown phases of binary black hole coalescences”. In: *Phys. Rev. D* 103 (12 2021), p. 123023. DOI: 10.1103/PhysRevD.103.123023. URL: <https://link.aps.org/doi/10.1103/PhysRevD.103.123023>.
- [390] Yinpeng Dong et al. “Improving Interpretability of Deep Neural Networks with Semantic Information”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 975–983. DOI: 10.1109/CVPR.2017.110.
- [391] Quanshi Zhang et al. “Interpretable Convolutional Neural Networks”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8827–8836. DOI: 10.1109/CVPR.2018.00920.
- [392] Yu Zhang et al. “A Survey on Neural Network Interpretability”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742. DOI: 10.1109/TETCI.2021.3100641.
- [393] Kohitij Kar et al. “Interpretability of artificial neural network models in artificial intelligence versus neuroscience”. In: *Nature Machine Intelligence* 4.12 (Dec. 2022), pp. 1065–1067. DOI: 10.1038/s42256-022-00592-3. URL: <https://doi.org/10.1038/s42256-022-00592-3>.
- [394] Chris Olah et al. “Feature Visualization”. In: *Distill* 2.11 (Nov. 2017). DOI: 10.23915/distill.00007. URL: <https://doi.org/10.23915/distill.00007>.
- [395] Yi han Sheu. “Illuminating the Black Box: Interpreting Deep Neural Network Models for Psychiatric Research”. In: *Frontiers in Psychiatry* 11 (Oct. 2020). DOI: 10.3389/fpsy.2020.551299. URL: <https://doi.org/10.3389/fpsy.2020.551299>.

- [396] Mohammadtaher Safarzadeh et al. “Interpreting a Machine Learning Model for Detecting Gravitational Waves”. In: (Feb. 2022). arXiv: 2202.07399 [gr-qc].
- [397] Asad Khan et al. “Interpretable AI forecasting for numerical relativity waveforms of quasicircular, spinning, nonprecessing binary black hole mergers”. In: *Phys. Rev. D* 105 (2 2022), p. 024024. DOI: 10.1103/PhysRevD.105.024024. URL: <https://link.aps.org/doi/10.1103/PhysRevD.105.024024>.

Acknowledgements

The author acknowledges support by the Michael Stifel Center Jena within the project “A Virtual Werkstatt for Digitization in the Sciences” and thanks the Carl Zeiss Foundation for the financial support within the scope of the program line “Breakthroughs”, as well as the Theoretisch-Physikalisches Institut of the Friedrich-Schiller-Universität Jena. The computational experiments were performed on

- hardware provided by the TPI,
- the ARA cluster at the Friedrich-Schiller-Universität Jena supported in part by the DFG grants INST 275/334-1 FUGG and INST 275/363-1 FUGG and ERC Starting Grant, no. BinGraSp-714626,
- the Atlas cluster financed by the Gottfried Wilhelm Leibniz Universität Hannover and the Max-Planck-Gesellschaft through the Albert-Einstein-Institut Hannover. I would like to thank the Observational Relativity and Cosmology division and its director prof. Bruce Allen for granting me access.

I am grateful to the COST network CA17137 “G2net” for support.

This research has made use of data or software obtained from the Gravitational Wave Open Science Center (gwosc.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. KAGRA is supported by Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan Society for the Promotion of Science (JSPS) in Japan; National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea; Academia Sinica (AS) and National Science and Technology Council (NSTC) in Taiwan.

I am very grateful to my supervisor prof. Bernd Brüggmann and all his assistance in the research described in this thesis, as well as Marlin Schäfer, Alex Nitz and Frank Ohme, who have personally taken part in the projects covered in this thesis. I thank all members of teams which submitted their contributions to the MLGWSC-1, as well as scientific advisors Elena Cuoco, Eliu Huerta, and Chris Messenger.

I thank Matteo Breschi for invaluable discussions about data processing and conventional analyses and Marlin Schäfer and Rossella Gamba for their helpful notes on this manuscript. In addition, I am grateful to Georgios Lukes-Gerakopoulos and Vojtěch Witzany, who have provided me with my initial research training and experience. Finally, I thank my family for unconditional support before, during, and after the writing of this dissertation.

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen unentgeltlich geholfen:

1. Prof. Bernd Brüggemann
2. Prof. Frank Ohme
3. Dr. Alexander Harvey Nitz
4. Dr. Marlin Benedikt Schäfer

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen.

Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die geltende Promotionsordnung der Physikalisch-Astronomischen Fakultät ist mir bekannt.

Ich versichere ehrenwörtlich, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Ort, Datum

Unterschrift d. Verfassers