Studies on MACHINE-LEARNING BASED DETECTION OF OVERLAPPING GRAVITATIONAL-WAVE SIGNALS IN SYNTHETIC EINSTEIN TELESCOPE DATA

A THESIS

Submitted In Partial Fulfillment
of the Requirements for the Degree of
MASTER OF SCIENCE

in the
INSTITUTE OF NUCLEAR PHYSICS

at
UNIVERSITÄT MÜNSTER

by
Sebastian WEIN

Supervised by
Prof. Dr. A. KAPPES
and Co-Examined by
Prof. Dr. A. ANDRONIC



Declaration of Academic Integrity

I hereby confirm that this thesis, entitled Studies on Machine-Learning Based Detection of Overlapping Gravitational-Wave Signals in Synthetic Einstein Telescope Data is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media and AI Tools, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited. I am aware that plagiarism is considered an act of deception which can result in sanction in accordance with the examination regulations.

27.7.2026, 

(date, signature of student)

I consent to having my thesis cross-checked with other texts to identify possible similarities and to having it stored in a database for this purpose.

I confirm that I have not submitted the following thesis in part or whole as an examination paper before.

27.7.2026, 

(date, signature of student)

Contents

1 Introduction **1**

2 Gravitational Waves **3**
2.1 General Relativity . . . 3
2.2 Properties in Vacuum . . . 4
2.3 Production . . . 6
2.4 Compact Binary Coalescences . . . 7

3 The Einstein Telescope **9**
3.1 Interferometric Detection . . . 9
3.2 Conceptual Design . . . 12
3.3 Scientific Goals . . . 14

4 Low-Latency Searches **15**
4.1 Problem Setup . . . 15
4.2 Matched Filtering . . . 16
4.3 Challenges and Thesis Motivation . . . 17

5 Machine Learning **19**
5.1 Basics . . . 20
5.2 Convolutional Neural Networks . . . 22
5.3 Transformers . . . 24

6 Data Simulation **31**
6.1 Detector Noise . . . 31
6.2 Gravitational Wave Signals . . . 34
6.3 Samples and Datasets . . . 38
6.4 Post-Processing . . . 40

7 Binary Classification **43**
7.1 Problem Setup . . . 43
7.2 CNN and Transformer Models . . . 44
7.3 Training . . . 47
7.4 Performance Comparison . . . 48
7.5 Parameter Biases . . . 48
7.6 Attention Analysis . . . 52
7.7 Summary . . . 54

8 Heatmap Regression **55**
8.1 Problem Setup 55
8.2 Heatmap Construction and Post-Processing 57
8.3 CNN and Transformer Models 59
8.4 Performance Comparison 61
8.5 Injection Study 64
8.6 Summary 68

9 Detection Transformer **69**
9.1 Architecture 69
9.2 Training 71
9.3 Comparison with Heatmap Regression 73
9.4 Injection Study 76
9.5 Discussion 78

10 Conclusion **79**

A Quadrupole Moment **87**

1 Introduction

Gravitational waves (GWs) are small oscillating perturbations of the spacetime metric caused by accelerating masses. Because they are so faint, only GW signals emitted by massive astrophysical objects have any chance of being detected. In particular, these are GWs emitted during the coalescence, *i.e.* the inspiral and merger, of a system of two black holes or neutron stars. These types of systems are called *compact binary coalescences* (CBCs) and give rise to a "chirp"-like GW signal, increasing in frequency and amplitude up until the merger of the two objects.

Large-scale laser interferometers allow for the detection of these GW signals. They are used at facilities such as LIGO [1, 2], where the first direct observation of gravitational waves was achieved in 2015 [3]. Currently there are a handful of such sites operating, forming the network of second-generation GW detectors.

As the current generation of detectors is reaching their design sensitivities and facility limits, a new third generation of detectors is being proposed. One of these is the Einstein Telescope (ET) [4, 5], which will be based in Europe. ET aims for an improved design sensitivity compared to second-generation detectors, particularly in the low-frequency regime $\mathcal{O}(1\,\mathrm{Hz})$. However, with this increase in sensitivity, new challenges for *signal detection* arise. In this context, signal detection refers to the identification of GW signals in the detector data stream. The challenge is mostly the growing quantity of information in the data and the increased computational power needed to extract it. ET will detect more signals in total, and these signals will stay up to minutes or hours in the detector's sensitivity band. Combined, this leads to frequent signal time-overlap.

For second-generation data, GW searches rely on *matched filtering* to produce event candidates [6, 7]. However, this algorithm is known to not scale well in time and memory complexity when handling long and overlapping signals. This is especially problematic for *low-latency* searches, which produce triggers for data monitoring and multi-messenger astronomy in real time.

Previously, the need for low-latency search algorithms has motivated the investigation of alternatives to matched filtering. Notably, *machine-learning* has shown promising results while being far more computationally efficient [8, 9]. The research focuses mostly on convolutional neural networks (CNNs) for detecting GW signals in second-generation detector conditions. For ET-like data with overlapping signals, a different type of architecture has proven successful for GW parameter estimation instead [10]. Namely, that is the Transformer [11] architecture, whose performance for GW detection is not well studied.

This work aims to implement machine-learning models to detect long-duration, overlapping CBC signals in ET conditions. Motivated by the results in [10], the performance of transformer-based architectures will be compared to more standard CNN approaches.

The rest of this thesis is structured as follows: Ch. 2 provides some theoretical background on GWs, and Ch. 3 introduces ET. Ch. 4 describes GW searches with matched filtering. After that follows an introduction to machine learning in Ch. 5, focusing on CNNs and the

Transformer. Training neural networks requires synthetic ET-like data, and Ch. 6 will be a detailed description of the simulation of this data.

The results are split in two parts. Ch. 7 isolates the problem of processing the large inputs required to capture the long-duration signals that are expected for ET. Chs. 8 and 9 deal with detecting multiple overlapping signals. Each time, two models are compared: one more closely resembling existing approaches and one being novel. The thesis concludes with a summary of its findings and an outlook in Ch. 10.

2 Gravitational Waves

The existence of gravitational waves is a direct consequence of Einstein's theory of general relativity (GR), which describes gravity in terms of the differential geometry of curved spacetime. The theory states that the presence of matter causes a curvature of spacetime and particles move along the shortest path in this curved space, along so-called *geodesics*. This chapter provides a brief introduction to general relativity and how gravitational waves emerge from this theory. It discusses their properties and astrophysical sources, focusing on compact binary systems since these are the only type of signals considered in this work. The sources for this chapter are [12] and to a lesser extent [13].

2.1 General Relativity

The theory of general relativity is based on the *strong equivalence principle*, which states that there is no way to distinguish free-falling frames from one another by performing local experiments and that locally, *special relativity* holds. This postulate alone gives rise to effects such as gravitational redshift or the bending of light near large masses. The equivalence principle then leads to a four-dimensional geometric description of spacetime, which is based on the metric tensor $g_{\mu\nu}$ whose indices denote the time and spatial axes, t, x, y and z. The metric tensor defines local distances in curved spacetime which are invariant under coordinate transformations through the differential line element,

$$\mathrm{d}s^2 = g_{\mu\nu}\mathrm{d}x^\mu \mathrm{d}x^\nu \tag{2.1}$$

with $x^\mu = (ct, \vec{r})$ where c is the speed of light. This is a generalization from the invariant quadratic form in special relativity, where spacetime is flat. Then, the metric tensor is the *Minkowski* metric $\eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$ and Eq. 2.1 reads

$$\mathrm{d}s^2 = -c^2\mathrm{d}t^2 + \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2. \tag{2.2}$$

The equations of motion for the metric $g_{\mu\nu}$ can be derived by defining the gravitational action, which depends on the metric itself and the matter content. Then, by requiring that the variation of this action with respect to the metric be zero, one arrives at the *Einstein equations*,

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}. \tag{2.3}$$

These equate the *energy-momentum tensor* $T_{\mu\nu}$ with the *Einstein tensor* $G_{\mu\nu} = R_{\mu\nu} - Rg_{\mu\nu}/2$ on the left-hand side up to some constant, where G is Newton's constant. The energy-momentum tensor describes the matter and energy content and the Einstein tensor expresses the resulting spacetime curvature. The latter is constructed from the metric $g_{\mu\nu}$ as well as the *Ricci tensor*

$R_{\mu\nu}$ and the *Ricci scalar* R, both of which also depend on the metric through various partial derivatives and index contractions.

Without any forces acting, particle trajectories will follow geodesics of this curved spacetime. Knowing the spacetime curvature, these can be calculated using the *geodesic equation*,

$$\frac{\mathrm{d}^2 x^\lambda}{\mathrm{d}\tau^2} + \Gamma^\lambda_{\mu\nu} \frac{\mathrm{d}x^\mu}{\mathrm{d}\tau} \frac{\mathrm{d}x^\nu}{\mathrm{d}\tau} = 0 \tag{2.4}$$

where τ is the particle's proper time and $\Gamma^\lambda_{\mu\nu}$ denote the *Christoffel symbols* which depend on the curvature.

It can be seen that GWs naturally emerge from GR when working in *linearized theory*: The GW's effect on spacetime is thought of as a "small" perturbation $h_{\mu\nu}$ of flat spacetime. Then, spacetime can be expanded as

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \quad \text{with } |h_{\mu\nu}| \ll 1, \tag{2.5}$$

and any terms of higher order than $\mathcal{O}(h_{\mu\nu})$ are neglected in following calculations. Further simplifications can be made by noting that GR is invariant under *all possible* coordinate transformations. This corresponds to large number of degrees of freedom, *i.e.*, a large *gauge symmetry*. Imposing linearized theory breaks that invariance, but still a residual gauge symmetry remains. Thus, the *Lorenz gauge* can be chosen, $\partial^\mu \bar{h}_{\mu\nu} = 0$. Then the Einstein equations (Eq. 2.3) simply reduce to wave equations,

$$\Box\, \bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4} T_{\mu\nu} \tag{2.6}$$

with the d'Alembertian $\Box = \partial_\mu \partial^\mu = -\partial_t^2/c^2 + \nabla^2$.[1] Here, $\bar{h}_{\mu\nu}$ denotes the *trace-reversed* perturbation, which is defined as

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu} h, \tag{2.7}$$

where $h = \eta^{\mu\nu} h_{\mu\nu}$ is the original perturbation's trace. The trace-reversed perturbation gets its name from the fact that its trace $\bar{h} = \eta^{\mu\nu} \bar{h}_{\mu\nu} = -h$. Obviously, the original perturbation can be reconstructed from the trace-reversed one, so no information is lost. Eq. 2.6 shows that GWs obey an inhomogeneous wave equation with the matter content $T_{\mu\nu}$ as its source.

2.2 Properties in Vacuum

To study the propagation of GWs and their influence on nearby particles (also called test masses in this context) it makes sense to consider the case far away from the sources that produce them. This is called the *vacuum case*, where $T_{\mu\nu} = 0$. The Einstein equations (Eq. 2.6) now read,

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

These equations have plane-wave solutions,

$$\bar{h}_{\mu\nu} = A_{\mu\nu}\, \mathrm{e}^{i k_\rho x^\rho} \tag{2.9}$$

with a wave-vector $k^\mu = (\omega/c, \vec{k})$ and some constant amplitude $A_{\mu\nu}$. This introduces complex numbers, and only the real part is considered at the end of calculations. For the plane-wave

[1] Indices are raised and lowered using the Minkowski metric in linearized theory.

solutions, the Einstein equations (Eq. 2.8) imply that $k_\mu k^\mu = 0$, which loosely translates to the fact that GWs propagate at the speed of light. The Lorenz gauge condition yields $A^{\mu\nu} k_\mu = 0$, *i.e.*, the amplitude and wave-vector are orthogonal.

Now for reasons that will not be laid out here, the vacuum case allows for the introduction of the *transverse-traceless* gauge with

$$h^{0\mu} = 0 \quad \text{and} \quad h^i{}_i = 0, \tag{2.10}$$

where Latin indices, such as i, j, k, denote spatial components only. For further information, consult [12]. In this gauge the trace h vanishes and thus $\bar{h}_{\mu\nu} = h_{\mu\nu}$. The Lorenz condition then reads $A^{ij} k_j = 0$: GWs are transverse, meaning they perturb the metric orthogonal to their propagation direction.

To illustrate their effect on test masses, one can arbitrarily choose a GW to propagate along the z-axis, $\vec{k} = (0, 0, \omega/c)$. Enforcing the gauge conditions, only two components of the amplitude remain and the GW perturbation takes the following form:

$$h_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A_+ & A_\times & 0 \\ 0 & A_\times & -A_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{\mu\nu} \mathrm{e}^{i\omega(t-\frac{z}{c})} = h_+ e^+_{\mu\nu} + h_\times e^\times_{\mu\nu} \tag{2.11}$$

with $h_+ = A_+ \exp i\omega(t - z/c)$ and $h_\times = A_\times \exp i\omega(t - z/c)$. The result is split along two linearly independent polarizations, labeled *plus* (+) and *cross* (×). Each is described by a polarization tensor, $e^+_{\mu\nu}$ and $e^\times_{\mu\nu}$, which are orthogonal, $e^+_{\mu\nu} e^{\times\mu\nu} = 0$. The plus and cross polarization states get their names from their effect on nearby test masses.

The trajectories of nearby particles with separation vectors S^μ follow from the geodesic equation (2.4). It can be seen that the test masses are only perturbed perpendicular to the GW's propagation direction in the x–y plane. In particular, solving for a purely plus-polarized wave, $h_\times = 0$, yields

$$S^1 = S^1(0) + \frac{h_+}{2} S^1(0), \quad S^2 = S^2(0) - \frac{h_+}{2} S^2(0). \tag{2.12}$$

Thus, particles initially separated on the x-axis will oscillate in that same direction; likewise for particles on the y-axis but with a phase shift of 180°. Considering a ring of test particles

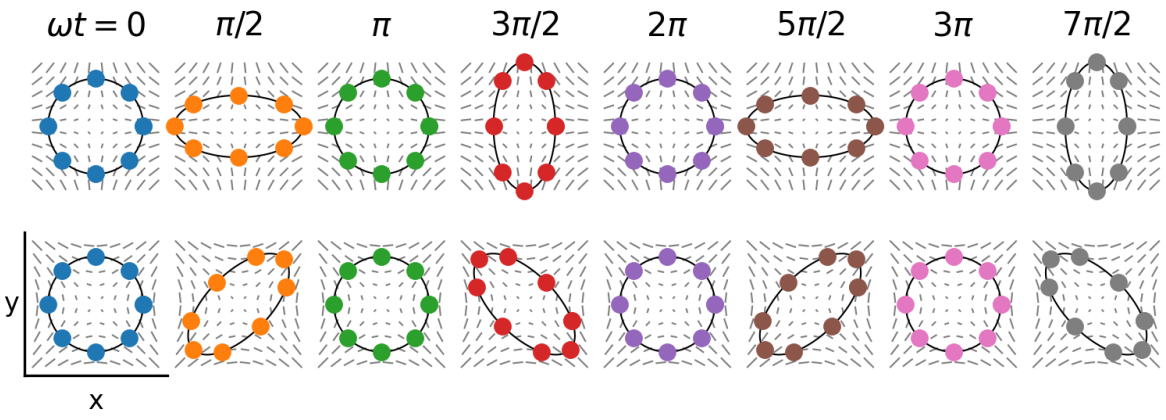


Figure 2.1: The effect of a passing GW on a ring of test masses on the transverse plane. The top row shows a purely plus-polarized GW, and the bottom row a cross-polarized one.

around the origin, one can see that they oscillate in a plus-shaped manner. Now, in the case that the GW is purely cross-polarized, $h_+ = 0$ and $h_\times \neq 0$, the solution is instead given by

$$S^1 = S^1(0) + \frac{h_\times}{2} S^2(0), \quad S^2 \ = S^2(0) - \frac{h_\times}{2} S^1(0), \tag{2.13}$$

and a ring of test masses will oscillate cross-wise. The effect of these two polarization states can be seen plotted in Fig. 2.1.

2.3 Production

Having studied their properties in the vacuum case, one can now take one step back and consider the production of GWs. This means considering $T_{\mu\nu} \neq 0$ (where now only the Lorenz gauge and *not* the transverse-traceless gauge can be chosen), in which case the Einstein equations take the form of an inhomogeneous wave equation again, $\Box \bar{h}_{\mu\nu} = -(16\pi G/c^4)\, T_{\mu\nu}$ (see Eq. 2.6). The solutions to this equation are known:

$$\bar{h}_{\mu\nu} = \frac{4G}{c^4} \int \mathrm{d}^3\vec{r}' \, \frac{T_{\mu\nu}\left(t - \frac{|\vec{r}-\vec{r}'|}{c}, \vec{r}'\right)}{|\vec{r} - \vec{r}'|}. \tag{2.14}$$

That is, the GW perturbation observed at position $(t, \vec{r})$ is caused by the sum of all sources $T_{\mu\nu}$ evaluated at time $t_{\mathrm{ret}} = t - |\vec{r} - \vec{r}'|/c$, the *retarded time*. (If a source at $\vec{r}'$ emits a signal at time t_{ret} it will reach the observer position $\vec{r}$ at time t.) Making the assumption that the source volume is "small" and that it is far away from the observer allows approximating the distance between the two as $|\vec{r} - \vec{r}'| \approx |\vec{r}| = r$. Thus,

$$\bar{h}_{\mu\nu} = \frac{4G}{c^4 r} \int \mathrm{d}^3\vec{r}' \, T_{\mu\nu}(t_{\mathrm{ret}}, \vec{r}'), \tag{2.15}$$

where the approximation is made that the source slowly moving such that $t_{\mathrm{ret}} = t - r/c$. Furthermore, properties of the stress-energy tensor turn out to prove that

$$\int \mathrm{d}^3\vec{r}\, T_{ij}(t, \vec{r}) = \frac{1}{2}\frac{\partial_t^2}{c^2} \int \mathrm{d}^3\vec{r}\, r_i r_j T_{00}(t, \vec{r}) = \frac{1}{2}\partial_t^2 Q_{ij}(t), \tag{2.16}$$

where Q_{ij} is the *quadrupole moment tensor*, defined as

$$Q_{ij}(t) = \frac{1}{c^2} \int \mathrm{d}^3\vec{r}\, r_i r_j T_{00}(t, \vec{r}). \tag{2.17}$$

Putting all together, the spatial components of the trace-reversed perturbation are given by

$$\bar{h}_{ij} = \frac{2G}{c^4}\frac{1}{r}\partial_t^2 Q_{ij}(t_{\mathrm{ret}}), \tag{2.18}$$

and the remaining four components $\bar{h}^{0\mu}$ can be found by demanding the Lorenz gauge condition be satisfied.

Eq. 2.18 is known as the *quadrupole formula*. It implies two things: that GWs are emitted by systems with a time-varying quadrupole moment (to leading order) and that their amplitude decays inversely with the distance from the source. Thus, static sources do not radiate and

neither do spherically symmetric sources since their mass quadrupole moment is zero[2]. Note that the prefactor G/c^4 is of order 10^{-45} in SI units, so GW perturbations are typically very small. The most promising sources for GW production will be massive objects with large kinetic energy. Such sources include rapidly spinning neutron stars with some mass asymmetry or compact binary systems of black holes (BBH) or neutron stars (BNS).

2.4 Compact Binary Coalescences

This latter case of radiating compact binary systems is called *compact binary coalescences* (CBCs). These are systems of two masses in compact orbit. The system loses energy in the form of GW emission, thus the orbital radius will decrease until the two objects merge. This is the only type of GW signal detected so far, which is mostly emitted by black hole binaries.

Using the quadrupole formula, the approximate form of the emitted GW can be calculated. (This closely follows [12].) To this end, the orbital motion is assumed to be Newtonian and the bodies treated as point-particles. The masses will be denoted m_1 and m_2 and their positions $\vec{r}_1$ and $\vec{r}_2$. In the center-of-mass frame, the system reduces to a one-body problem with effective mass $\mu = m_1 m_2/M$ and relative position $\vec{\rho} = \vec{r}_2 - \vec{r}_1$, where $M = m_1 + m_2$ is the total mass. The bodies' motion is governed by $\partial_t^2 \vec{\rho} = -(GM/R^3)\,\vec{\rho}$ with $R = |\vec{\rho}|$. In the simplest case, the orbit will be circular with angular frequency $\omega = \sqrt{GM/R^3}$. Choosing the x–y plane as the orbital plane, this means

$$\vec{\rho} = R\,\left(\cos\!\left(\omega t + \tfrac{\pi}{2}\right) \quad \sin\!\left(\omega t + \tfrac{\pi}{2}\right) \quad 0\right)^{\mathsf{T}}. \tag{2.19}$$

Then, $T_{00}(t,\vec{r}) = c^2\mu\delta(\vec{r} - \vec{\rho})$, and after projecting the quadrupole moment tensor to the transverse-traceless gauge, the quadrupole formula yields

$$\begin{aligned} h_+(t) &= \frac{4}{r}\left(\frac{G\mathcal{M}}{c^2}\right)^{\frac{5}{3}}\left(\frac{\omega}{c}\right)^{\frac{2}{3}}\frac{1+\cos^2\theta}{2}\,\cos(2\omega t_{\mathrm{ret}} + 2\phi), \\ h_\times(t) &= \frac{4}{r}\left(\frac{G\mathcal{M}}{c^2}\right)^{\frac{5}{3}}\left(\frac{\omega}{c}\right)^{\frac{2}{3}}\cos\theta\,\sin(2\omega t_{\mathrm{ret}} + 2\phi), \end{aligned} \tag{2.20}$$

where θ and ϕ are the polar and azimuth angles respectively, and the *chirp mass* is introduced as $\mathcal{M} = \mu^{3/5}(m_1 + m_2)^{2/5}$. Note that the resulting GW signal oscillates with a frequency f that is twice the orbital frequency $\nu = \omega/2\pi$, that is, $f = 2\nu = \omega/\pi$.

It can be shown that the radiated power per unit solid angle takes on the following form:

$$\frac{\mathrm{d}P}{\mathrm{d}\Omega} = \frac{2}{\pi}\frac{c^5}{G}\left(\frac{G\mathcal{M}\omega}{c^3}\right)^{\frac{10}{3}} g(\theta) \tag{2.21}$$

with an angular distribution $g(\theta) = (1+\cos^2\theta)^2/4 + \cos^2\theta$. (For further information consult [12].) This reveals that the radiated power scales with the orbital angular frequency $\omega^{10/3}$ and that most of that power is radiated along the z-axis, *i.e.* perpendicular to the orbital plane (see Fig. 2.2). Now, the dependence on the frequency creates this feedback loop where the system continuously loses energy, which in turn decreases the orbital radius and increases the frequency leading to higher energy loss. This process gives rise to the characteristic chirp-like signal that is associated with CBCs, where up until the merger, the GW frequency and amplitude increase.

[2]The quadrupole moment is only non-vanishing for mass distributions with a non-zero $l = 2$ spherical harmonic contribution. Thus spherically symmetric systems ($l = 0$) do not emit GWs, while orbiting binary systems ($l = 2$) do radiate. See App. A

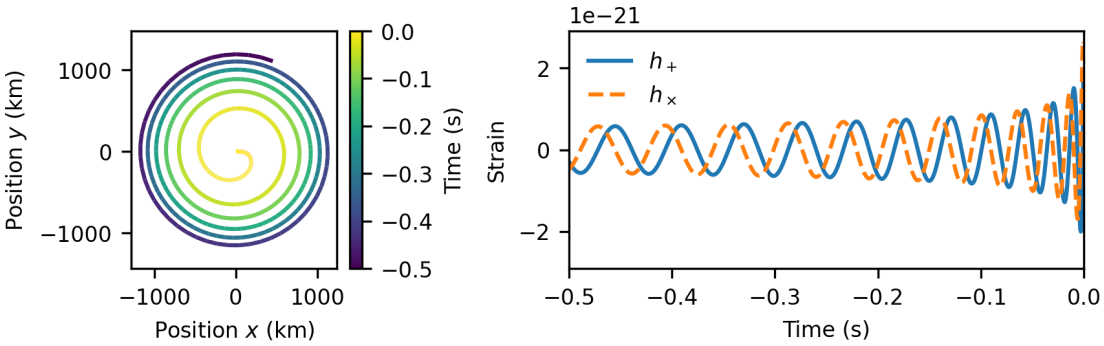


Figure 2.3: Evolution of a CBC system. Trajectory of the reduced mass in the orbital plane in the center-of-mass frame on the left. The two polarizations of the emitted GW on the right.

The evolution of $\omega(t)$ can be calculated by considering that the total energy of this system is $E = -Gm_1m_2/2R$. Equating P and $-\mathrm{d}E/\mathrm{d}t$ results in a differential equation for ω. Solving with respect to $\tau = t_{\mathrm{coa}} - t$, the time to coalesce, yields

$$\omega(\tau) = \left(\frac{c^3}{G\mathcal{M}}\right)^{\frac{5}{8}} \left(\frac{5}{256}\frac{1}{\tau}\right)^{\frac{3}{8}}. \tag{2.22}$$

With this result, the one-body motion in the center-of-mass frame can be obtained via Eq. 2.19 as well as the resulting GW waveform from Eq. 2.20. The dynamics of two inspiraling masses, $40\,\mathrm{M}_\odot$ and $30\,\mathrm{M}_\odot$, and the emitted GW signal as measured at a distance of $r = 500\,\mathrm{Mpc}$ are plotted in Fig. 2.3. It can be seen in the last 0.5 s, the objects are very compact being separated by $\mathcal{O}(100\,\mathrm{km})$. The corresponding signal is chirp-like and of order $\mathcal{O}(10^{-21})$, so very faint despite the large masses.

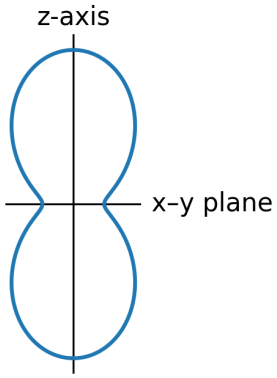


Figure 2.2: Radiation pattern of a CBC system in the x–y plane visualized by plotting $g(\theta)\,(\cos\theta,\,\sin\theta)$

Observe that formally, the frequency diverges at the time of coalescence (Eq. 2.22), which is of course unphysical. This divergence stems from the fact that the two objects get arbitrarily close. Realistically, the innermost stable orbit would be $r_{\mathrm{ISCO}} = 6GM/c^2$, which corresponds to an angular orbital frequency of $\omega_{\mathrm{ISCO}} = c^3/6\sqrt{6}GM$. This can be related to the emitted GW signal frequency as

$$f_{\mathrm{max}} = \frac{\omega_{\mathrm{ISCO}}}{\pi} \approx 4.4\,\mathrm{kHz}\,\frac{1\,\mathrm{M}_\odot}{M}. \tag{2.23}$$

For black hole binaries with masses $\mathcal{O}(10\,\mathrm{M}_\odot)$ this means maximum frequencies of $\mathcal{O}(100\,\mathrm{Hz})$ while neutron star binaries have masses $\approx 1\,\mathrm{M}_\odot$ resulting in frequencies up to $\approx 2000\,\mathrm{Hz}$.

3 The Einstein Telescope

The use of laser interferometers allows for the detection of gravitational waves at sites such as LIGO [1, 2], Virgo [14, 15] and KAGRA [16]. These form the current second generation of GW observatories, which have opened the field of gravitational wave (GW) astronomy. The second-generation detectors are, however, approaching their design sensitivities, with future upgrades reaching the facility limits. As many science cases require an improvement in sensitivity by roughly one order of magnitude, a new third generation of detectors is being proposed. One of these is the Einstein Telescope (ET) [4, 5], which will be based in Europe. This chapter introduces ET, outlining the detection principle, its conceptual design and scientific goals.

3.1 Interferometric Detection

Ch. 2.3 showed that the GW perturbations as measured on Earth are incredibly tiny, with amplitudes of $h \lesssim 10^{-21}$, posing a significant challenge for direct detection. GWs are measured directly by observing their effect on test masses. From Eqs. 2.12 and 2.13, it is clear that a passing GW will cause displacement of the same order of magnitude as the perturbation h. This effect can be exploited using a *Michelson laser interferometer* with its mirrors serving as the test masses.

In the simplest case, this instrument consists of one laser, a beam splitter, two mirrors and a photo diode. The laser is directed towards the partially reflective beam splitter. The photons are sent down the two interferometer arms and reflected by mirrors at each end. The two beams travel back to the beam splitter where they are redirected to the photodiode measuring the resulting interference. A schematic is shown in Fig. 3.1. The instrument is calibrated such that, in the absence of a GW, the beams interfere destructively at the photodiode. This is called *dark fringe operation*.

This setup is enhanced by the inclusion of extra mirrors forming so-called *Fabry-Perot cavities* with those mirrors at the end of each arm (see Fig. 3.1). The cavities increase the effective arm length as photons bounce back and forth a few hundred times before reaching the beam splitter again.

Now, suppose a purely plus polarized GW passes along the z-direction, and interferometer's arms lie on the x-axis and the y-axis. If the arms are of length L, the GW will cause length changes of $\Delta L_x = h_+ L/2$ and $\Delta L_y = -h_+ L/2$ respectively (see Eq. 2.12). This leads to a path difference

$$\Delta L = \Delta L_x - \Delta L_y = h_+ L, \tag{3.1}$$

which results in a phase difference of $\Delta\phi = (2\pi/\lambda)\Delta L$ with the laser wavelength λ. For arms of $\mathcal{O}(1\,\mathrm{km})$ the path differences will be $\mathcal{O}(10^{-18}\,\mathrm{m})$, which is much smaller than typical laser wavelengths of $\mathcal{O}(10^{-6}\,\mathrm{m})$. Thus, phase difference produces a small deviation from destructive interference and the measured photodiode voltage will be proportional to the

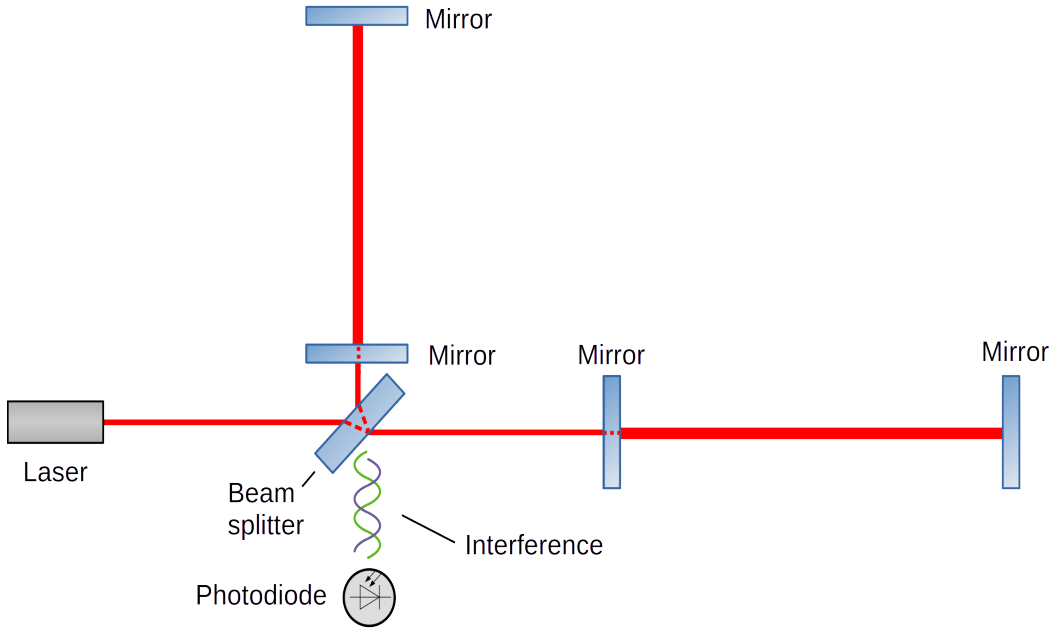


Figure 3.1: Michelson laser-interferometer schematic, not drawn to scale. Usually, both arms will be of equal length. The extra mirrors near the beam splitter form power-enhancing *Fabry-Perot cavities*.

path difference. After calibrating the detector, the photodiode output can be converted to path differences, which, according to Eq. 3.1, can be divided by the arm length to yield the perturbation h_+.

In fact, Eq. 3.1 can be generalized for GWs of any polarization and any propagation direction. The path length change divided by arm length will then be

$$\frac{\Delta L}{L} = F_+ h_+ + F_\times h_\times, \tag{3.2}$$

where F_+ and $F_\times$ are in $\mathcal{O}(1)$. This quantity is called the *strain*, which is the projection of the GW onto the detector, and serves as the data which is used for further analysis. In an act of notation abuse, the GW strain is denoted h, not to be confused with the trace of $h_{\mu\nu}$. F_+ and $F_\times$ are the *antenna pattern functions*. For a GW with polarization ψ and incoming direction given by θ and φ, the polar and azimuth angles, these read

$$\begin{aligned} F_+ &= \frac{\sin\gamma}{2}((1+\cos^2\theta)\cos 2\varphi\cos 2\psi - 2\cos\theta\sin 2\varphi\sin 2\psi), \\ F_\times &= \frac{\sin\gamma}{2}((1+\cos^2\theta)\cos 2\varphi\sin 2\psi + 2\cos\theta\sin 2\varphi\cos 2\psi). \end{aligned} \tag{3.3}$$

Here, the interferometer arm lies on the x–y plane with one arm on the x-axis while the other one is rotated by γ, which denotes the detector opening angle. Notice that there are four cases where both pattern functions vanish. These are the blind spots of the detector, which occur for GWs that propagate along the detector plane, *i.e.* $\theta = \pi/2$. Then the interferometer is blind to those GWs with incident azimuth angles φ of 45°, 135°, 225° and 315°.

This detection principle is employed with great success at LIGO, which consists of two interferometers, H1 and L1, separated by 3000 km. These have arm lengths of 4 km and 90° opening angles. The two sites allow for the rejection of false signals via coincident

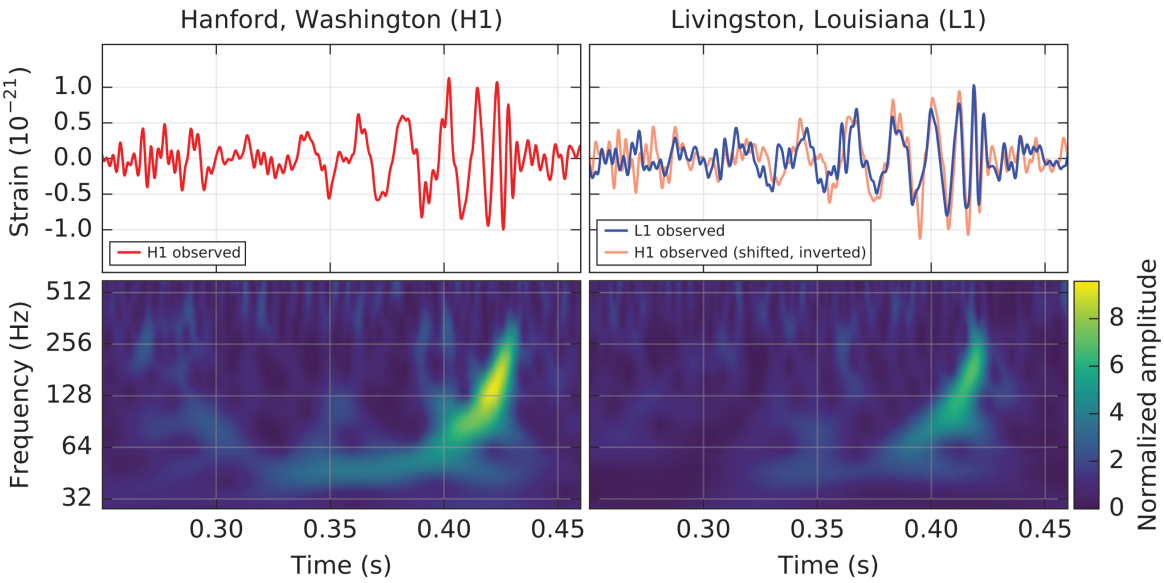


Figure 3.2: GW150914 as measured at the two LIGO sites, H1 (*left column*) and L1 (*right column*) shifted by 7 ms and inverted to make up for the detectors' relative position and orientation. Strain (*top row*) and spectogram (*bottom row*). The data is bandpassed from 35 Hz to 350 Hz for visualization. Adapted from [3].

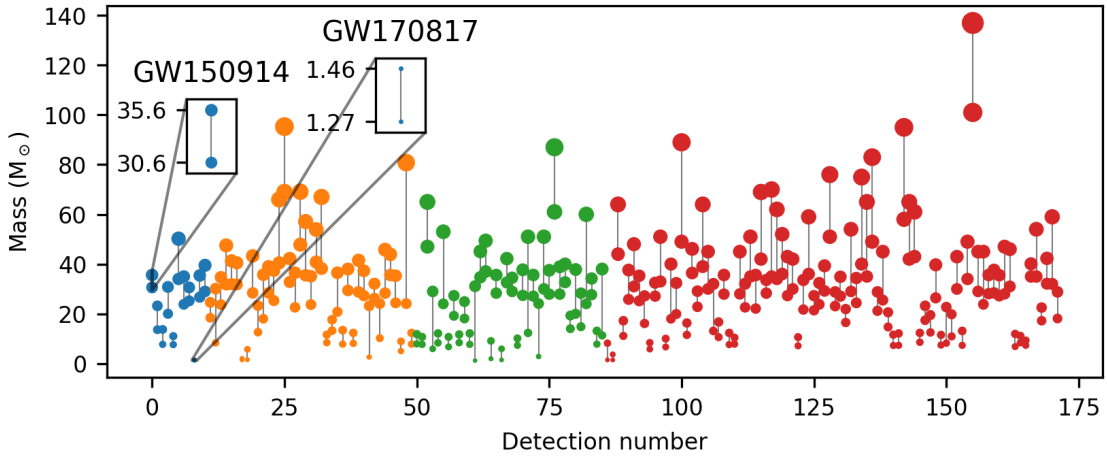


Figure 3.3: CBCs observed by LIGO, Virgo and KAGRA in their four observing runs, each plotted in a different color. Each merger is represented as two dots denoting the two component masses (in both y-value and dot size). Data from [17, 18, 19, 20], uncertainties are not plotted.

detection (< 10 ms, the time-of-flight between the two detectors). LIGO achieved the first direct detection of a GW signal on September 14th, 2015 [3] resulting in a Nobel prize award in 2017 [21]. The signal was identified as a BBH merger with masses $36^{+5}_{-4}\,M_\odot$ and $29^{+4}_{-4}\,M_\odot$ and with a luminosity distance of 410^{+160}_{-180} Mpc. GW signals are conventionally labeled based on their arrival time, so GW150914 in this case. See the measured signal in Fig. 3.2. So far, the LIGO, Virgo and KAGRA observatories have made more than 200 confident GW detections in the span of four observing runs [17, 18, 19, 20]. The majority of these are black hole mergers. BNS mergers are identified based on the component masses which have to be $\lesssim 2\,M_\odot$. The first direct detection of a neutron star merger was GW170817 [22]. See the signals for which component masses are provided in Fig. 3.3.

3.2 Conceptual Design

The Einstein Telescope (ET) aims for an increase in sensitivity of a factor ten compared to second-generation detectors, especially in the low frequency regime $\mathcal{O}(1\,\text{Hz})$. To this end, the arm length will be increased to 10 km yielding an overall sensitivity increase. To improve the low frequency sensitivity, where noise is mostly seismic, ET will be built 200 m to 300 m underground. A detailed description of the latest ET technical design is provided by [23].[1]

ET will consist of three nested detectors with 60° opening angles, arranged in a triangular pattern. This allows for better sky coverage at a slight sensitivity cost for interferometers with opening angles $\gamma < 90°$ (corresponding to the factor $\sin\gamma$ in Eqs. 3.3). The arrangement is equally sensitive to both GW polarizations. Each detector will comprise two interferometers, one optimized for low frequencies of 3 Hz to 30 Hz (ET-LF) and one for high frequencies 30 Hz to 10 kHz (ET-HF). This is called the *xylophone* configuration. See a schematic in Fig. 3.4.

This configuration is useful because the main noise contribution for these instruments, the *quantum noise*, behaves differently at low and high frequencies. This type of noise is caused by individual quanta of light, *i.e.* photons, and has two components. The *shot noise* results from the Poisson statistics of the total number of photons detected and dominates at high frequencies. The *radiation pressure noise* arises from the momentum transfer of individual photons to the mirrors upon reflection. Fluctuations in the photon number therefore induce random displacements of the mirrors. This noise dominates at low frequencies.

These two contributions scale oppositely with laser power: increasing the laser power reduces shot noise but increases radiation pressure. ET-HF mitigates shot noise by operating with a higher arm cavity power of 3 MW, while ET-LF suppresses radiation pressure by reducing the power to 18 kW. Compare this to the LIGO interferometers which balance both noise contributions by choosing 750 kW. In both ET-LF and ET-HF, radiation pressure can be further reduced by employing heavier test masses of 200 kg instead of 40 kg at LIGO.

Another noise source is *thermal noise* due to Brownian motion in the test masses and their suspensions. This limits the instrument's sensitivity in the low frequencies, hence ET-LF will operate at cryogenic temperatures of 10 K to 20 K. The poor low-temperature behavior of fused silica, which is used for the optics of ET-HF, necessitates employing silicon for the ET-LF test masses and suspensions. Since silicon is non-transparent to wavelengths of

[1] It should be mentioned that this technical report is rather old as of now. In particular, it focuses solely on a triangular configuration of three nested detectors with 60° opening angles. However, the detector geometry is currently debated with 2L configurations being investigated as well (two detectors with 90° opening angles) [24].

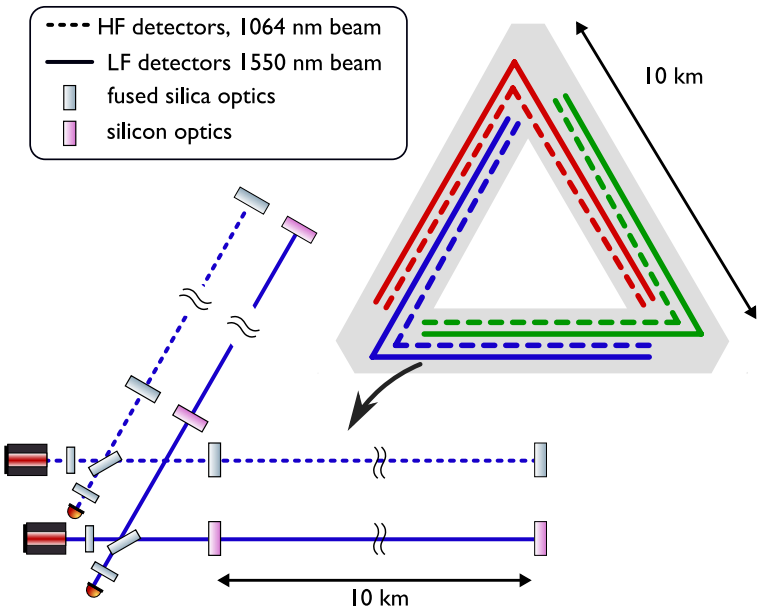


Figure 3.4: The ET xylophone design. Three detectors arranged in a triangular shape each consisting of a LF and HF interferometer. ET-LF will feature a 1550 nm laser wavelength and silicon optics, while ET-HF will employ a 1064 nm laser with fused silica optics, similar to LIGO and Virgo. Taken from [25].

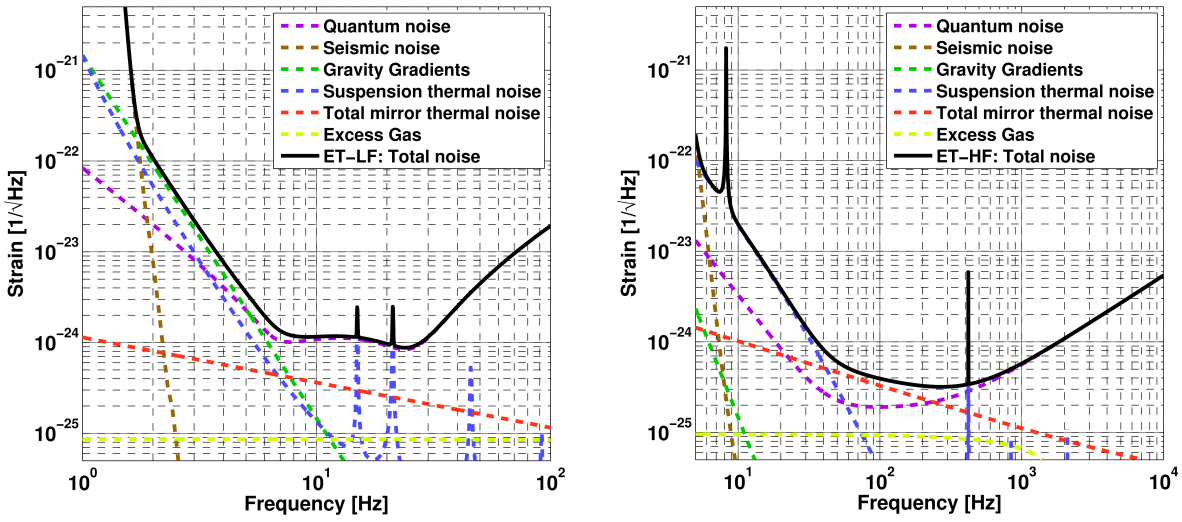


Figure 3.5: Noise budgets for the ET-LF (left) and ET-HF (right) interferometers. The individual sources (dotted) and the total noise taken as their quadrature sum (black). The label "Gravity gradients" denotes Newtonian noise. Excess gas noise stems from residual gas in the beam tubes. Taken from [5].

1064 nm, the standard for second-generation detectors, which will be used for ET-HF as well, ET-LF will move to a larger laser wavelength of 1550 nm.

As already mentioned, the instrument's low-frequency sensitivity is mainly limited by *seismic noise* and *Newtonian noise*. Both of these stem from seismic ground motion caused by local disturbances such as human activity or wind. Seismic noise describes the vibrations directly induced on the optics and can be shielded by a combination of active and passive isolation systems. ET will employ a multi-stage pendulum suspension system for its mirrors, similar to LIGO and Virgo.

Newtonian noise, on the other hand, describes disturbance of the local gravitational field caused by the mass density fluctuations associated with seismic ground motion and its effect on the test masses. Contrary to seismic noise, Newtonian noise cannot be shielded, which is the main argument for moving the detector 200 m to 300 m underground to a sparsely populated region. Candidates include the Euregio Meuse-Rhine and Sardinia [26].

ET is set to provide a sensitivity improvement of a factor ten, achieving a targeted sensitivity of $10^{-24}\,\mathrm{Hz}^{-1/2}$ and broadening the frequency range down to 3 Hz to 5 Hz, compared to the current detectors' 10 Hz to 20 Hz lower limit. See the ET noise budget for both interferometers in Fig. 3.5. Plotted here is the noise *amplitude spectral density* (ASD) which is the square root of the *power spectral density* (PSD) and is related to the root-mean-square noise amplitude at each frequency. (Ch. 4 will introduce these formally.)

3.3 Scientific Goals

With its increased sensitivity, ET will detect large numbers of CBCs up to high redshifts and with unprecedented precision. In numbers that is 10^5 to 10^6 BBH and 10^4 BNS signals annually [27]. This has the potential for advances in astrophysics, cosmology and fundamental physics. For a full discussion of ET's main science cases, see [23, 27].

ET will enable the study of the stellar black hole and neutron star populations, including their mass and spin distributions, their evolution with redshift, and their formation mechanisms. ET will probe black holes beyond the epoch of star formation, possibly uncovering black holes of primordial origin. These are hypothetical non-stellar black holes, formed in the high-density conditions of the early universe, which are proposed as a dark matter candidate.

High precision measurements of CBC signatures will provide accurate tests of GR, particularly in the extreme domain during the merger phase of BBHs. Similarly, high-precision BNS signatures allow for the study of quantum chromodynamics at ultra-high density, and of possible phase transitions in the neutron star core.

ET's excellent sky coverage and sensitivity make it capable of localizing GW sources with remarkable precision. This opens the opportunity for multi-messenger astronomy, where non-GW observatories probe the source's sky location for electromagnetic or neutrino follow-ups. This is especially important for mergers involving a neutron star, which can emit both GWs and electromagnetic radiation in a kilonova [28].

Besides CBC signals, ET may detect other types of GW sources. These include signals from core collapse supernovae, continuous signals emitted by isolated rotating neutron stars, and possibly the stochastic GW background.

4 Low-Latency Searches

GW signals are identified in detector data with search algorithms, returning (multiple) GW candidates. These are called *online* if they produce real-time or low-latency triggers on the data as they are taken and *offline* else. In the observing runs of the second-generation detectors, online searches were instrumental for data monitoring and multi-messenger astronomy [29]. These searches rely upon matched filtering to produce real-time triggers, and focused Bayesian inference for subsequent parameter estimation.

This chapter states the problem of identifying GW signals in the detector data formally. It introduces matched filtering, and its shortcomings for low-latency trigger generation in the high event-rate conditions of ET, motivating the investigation of alternative or complementary methods, *e.g.* machine-learning based. The chapter's main source is [12].

4.1 Problem Setup

For a given detector stream, a search algorithm will return a number of time triggers corresponding to identified GW events. The returned time will usually correspond to the CBC merger time and should be accompanied by some confidence measure.

The detector strain is a function of time $s(t)$, and is made up of two additive components, the instrument noise $n(t)$ and a possible signal contribution $h(t)$:

$$s(t) = n(t) + h(t). \tag{4.1}$$

For now, the signal contribution will be assumed to either vanish or correspond to a single CBC signal with merger time t_{coa}. The generalization to multiple signals will be explored later. If a signal is present, it will induce a strain according to Eq. 3.2 resulting in amplitudes of order 10^{-21} or smaller. Usually $|h(t)| \ll |n(t)|$, making the identification of this signal inside the much larger noise non-trivial.

The noise strain can be described as a statistical process, which is stationary, *i.e.*, its properties do not change over time. This process is characterized by the noise power spectral density $S_n(f)$ (PSD). The PSD is in turn related to the Fourier transform of the noise strain $\tilde{n}(f)$. In particular, through the expectation value $\langle\cdot\rangle$ of the Fourier components:

$$\langle \tilde{n}^*(f)\, \tilde{n}(f')\rangle = \delta(f - f')\, \frac{1}{2} S_n(f). \tag{4.2}$$

Here, δ denotes the Dirac delta function. Notice that since n is real, $\tilde{n}(-f)^* = \tilde{n}(f)$ and therefore $S_n(-f) = S_n(f)$. The factor $1/2$ is inserted such that the mean squared noise $\langle n(t)^2\rangle$

is obtained by integrating $S_n(f)$ over the physical range of $f \geq 0$, rather than from $-\infty$ to ∞:

$$\begin{aligned}\langle n(t)^2 \rangle &= \langle n^*(t)\, n(t) \rangle_{t=0} && (4.3)\\ &= \int_{-\infty}^{\infty} \mathrm{d}f \mathrm{d}f' \, \langle \tilde{n}^*(f) \tilde{n}(f') \rangle && (4.4)\\ &= \frac{1}{2} \int_{-\infty}^{\infty} \mathrm{d}f \, S_n(f) && (4.5)\\ &= \int_{0}^{\infty} \mathrm{d}f \, S_n(f), && (4.6)\end{aligned}$$

where Eq. 4.3 makes use of stationarity, and Eq. 4.5 follows from the PSD definition (Eq. 4.2). To emphasize this fact, the PSD is called *single-sided* if defined like this. This convention will be used in the rest of this work.

4.2 Matched Filtering

If the shape the signal strain can take on is known (as is the case for CBC signals), matched filtering can be applied. The basic idea is to "filter out" the noise contribution of the total detector strain by multiplying with a filter function and integrating over the observation time T. Although not necessarily optimal, this technique works for a filter that is equal to the signal:

$$\frac{1}{T} \int_0^T \mathrm{d}t s(t) h(t) = \frac{1}{T} \int_0^T \mathrm{d}t h^2(t) + \frac{1}{T} \int_0^T \mathrm{d}t n(t) h(t). \tag{4.7}$$

Here, the first integrand on the right-hand side is positive definite and the first summand will be constant for large T while the second summand which is the noise contribution will approach zero.

This procedure can be formalized by defining the filtered detector strain as

$$\hat{s} = \int_{-\infty}^{\infty} \mathrm{d}t \, s(t) K(t), \tag{4.8}$$

for some filter function $K(t)$. The *optimal* or *matched* filter should maximize the *signal-to-noise ratio* (SNR) of $\hat{s}$. The SNR is taken as the ratio of S, the expected value of $\hat{s}$ if a signal is present, and N, the root mean square of $\hat{s}$ if *no* signal is present. That is,

$$S = \langle \hat{s} \rangle \quad \text{and} \quad N^2 = [\langle \hat{s}^2 \rangle - \langle \hat{s} \rangle^2]_{h(t)=0}. \tag{4.9}$$

Requiring $\langle n(t) \rangle = 0$ and making use of the PSD definition (Eq. 4.2), the SNR takes the form,

$$\frac{S}{N} = \frac{\int_{-\infty}^{\infty} \mathrm{d}f \, \tilde{h}(f) \tilde{K}^*(f)}{\sqrt{\int_{-\infty}^{\infty} \mathrm{d}f \frac{1}{2} S_n(f) |\tilde{K}(f)|^2}}. \tag{4.10}$$

Skipping the derivation, the solution to the variational problem that is to find the optimal filter $K(t)$ maximizing S/N is given by,

$$\tilde{K}(f) \sim \frac{\tilde{h}(f)}{S_n(f)}. \tag{4.11}$$

See [12] for proof. Notice that for white noise $S_n(f) = 1$ and the optimal filter will be equal to the signal which corresponds to the naive approach taken earlier.

The search of the optimal filter is simply brute forced. To this end, a *template bank* of possible waveforms is constructed, covering the whole parameter space at a reasonable resolution. The template that yields the maximum SNR while exceeding some threshold will be the GW signal candidate. This is a strongly simplified search pipeline. A *full* pipeline is provided by the PyCBC software package [30, 7].

In practice, the expectation value $S = \langle \hat{s} \rangle$ and therefore the SNR, S/N, is obviously not available as it requires knowledge of the true GW signal $h(t)$. Instead, the SNR is calculated as $\rho = \hat{s}/N$, which depends on the full detector strain $s(t)$. This gives a "noisy" estimate of S/N, reflecting the instrument noise statistics. On average, $\langle \rho \rangle = S/N$.

Matched filtering will return a time trigger, *e.g.* the merger time of the template, and the corresponding SNR serves as a confidence measure. As a bonus, the optimal template gives a rough estimate of the signal parameters. The SNR corresponding to the optimal filter reads,

$$\left(\frac{S}{N}\right)_{\text{opt}} = 2\sqrt{\int_0^\infty \mathrm{d}f \, \frac{|\tilde{h}(f)|^2}{S_n(f)}}. \tag{4.12}$$

This is often and will also in this work be called the SNR of the signal $h(t)$ for a given noise PSD. Note that this is not literally the SNR of the signal $h(t)$ with respect to the detector noise $n(t)$, but rather the SNR of the optimally filtered detector strain $\hat{s}$. Still this wording is used because it gives a sense of the detectability or loudness of a signal. To give a sense of scale: for a random template the SNR will be 2 on average. The PyCBC matched filtering pipeline typically disregards templates with SNR values below a threshold of 5.5 [7].

4.3 Challenges and Thesis Motivation

As already mentioned, the high sensitivity of ET will introduce challenges for online GW searches with matched filtering. The problem is less an increase in data volume from the six interferometers (plus auxiliary channels) compared to the existing second-generation network. Rather, it is the quantity of valuable scientific information hidden in the data that will grow, and the increased computational power needed to extract it.

The problem is twofold: The increased sensitivity results in longer-duration signals, and a larger event-rate. Combined, this introduces the added challenge of signal overlap.

The increased CBC signal durations stem exclusively from ET's improvements in the low-frequency range. CBC signals are chirp-like, and will enter the ET sensitivity band earlier thus being detectable for longer. An approximation can be made by recalling that the binary's orbital frequency ω is related to the time to coalesce τ through Eq. 2.22, and that the GW signal frequency is $f = \omega/\pi$. Putting both together, and rearranging yields,

$$\tau \approx 2.18\,\text{s} \left(\frac{1.21\,\text{M}_\odot}{\mathcal{M}}\right)^{\frac{5}{3}} \left(\frac{100\,\text{Hz}}{f_{\text{lower}}}\right)^{\frac{8}{3}}, \tag{4.13}$$

the remaining signal duration once a frequency of f_{lower} is reached. GW150914 with chirp mass $30\,\text{M}_\odot$ entered LIGO's sensitivity band at 20 Hz and lasted for less than 1 s. In contrast, it would have entered ET's band around 4 Hz, lasting for a whole minute. This effect is even more pronounced for low-mass inspirals such as BNS mergers. GW170817 had a chirp mass of $1.2\,\text{M}_\odot$ and entered the LIGO band at 10 Hz, corresponding to a signal duration of 17 min. In ET that would be 1 Hz and total signal-duration of more than 24 h.

ET is expected to detect 10^5 to 10^6 CBC signals annually. As these events occur independently of one another, they can be assumed to follow a Poissonian distribution. For a

given signal rate λ, the probability that two events are separated by a time-duration of less than Δt is,

$$P(\Delta t) = 1 - \mathrm{e}^{-\lambda \Delta t}. \tag{4.14}$$

For the higher estimate of $\lambda = 10^6\,\mathrm{yr}^{-1}$, this means that the majority of all signals, 85 %, will be separated by less than a minute. A total of 3×10^4 signals will be separated by less than one second. Since for BBH signals (which make up the majority of mergers) durations of ~ 1 min are expected, this high merger rate introduces regular signal overlap.

These two effects introduce computational challenges for matched filtering. For one, the increased signal-duration necessitates longer waveform templates. This increases the template banks' memory needs, and slows down the algorithm, which is linear in time complexity with respect to template length. Secondly, the signal overlap, especially for mergers separated by less than one second, breaks the assumption that detector data contains, at most, one signal at any time.

These challenges have been addressed in [31]. Without adaption, matched filtering turns out to retain accuracy in *offline* GW searches when signals are separated by more than one second, but breaks down for closer signals. The latter case of separations of less than one second will make up a substantial amount of signals in ET. Also, it is not yet clear how *online* searches will perform in ET conditions. This motivates the investigation of alternative methods for low-latency GW searches.

This thesis studies the performance of machine-learning methods for low-latency searches in ET conditions. The scope of this work includes BBH signals only, while BNS mergers will be disregarded. In ET, the expected event rate of BBH mergers is significantly larger than for BNS. The use case of BBH low-latency searches is primarily data monitoring, as opposed to multi-messenger alerts. BNS signals complicate searches in two ways. Their signal-duration is much longer, and their potential for multi-messenger astronomy requires *negative*-latency alerts, where triggers are distributed prior to the signal's merger phase.

5 Machine Learning

Machine learning is concerned with the development of statistical algorithms that can "learn" from data and generalize to unseen data, without being explicitly programmed. One particular class of such algorithms is *neural networks*, originally inspired by the human brain, which have proven highly successful. Technically, these fall under *deep learning*, which is a subset of machine learning. Still, the use of deep-learning methods has become so ubiquitous (at least in physics applications) that both these terms are used more or less synonymously.

Machine learning requires large-scale matrix computations. These can be performed efficiently on graphics processing units (GPUs). This thesis makes use of *PyTorch* [32], which is Python-based, to handle these computations. PyTorch also serves as an extensive machine-learning library, which includes models, loss functions and optimizers. The hardware used in this thesis is provided by the HPC cluster PALMA II of the University of Münster.

The investigation of machine-learning methods to complement matched filtering, for low-latency searches in particular, is not novel [8]. Recently, the increased amount of data from the growing second-generation GW observatory network, and the corresponding computational burden has motivated such studies [9]. The research focuses primarily on convolutional neural networks (CNNs), and finds comparable performance while being far more computationally efficient, which is important for real-time detection.

This chapter introduces the basics of machine learning. This includes artificial neurons, how they are arranged into neural networks, and how these networks are trained to extract relevant information from data. Two architectures will be presented in detail, CNNs and the Transformer [11].

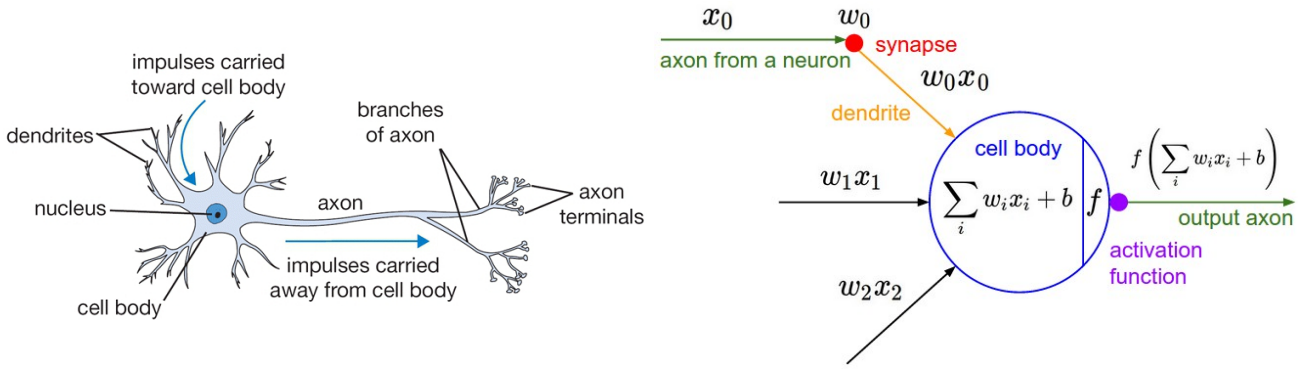


Figure 5.1: Drawing of a biological neuron (*left*) and the mathematical model of an artificial neuron (*right*). Taken from [33].

5.1 Basics

The fundamental building blocks of neural networks are *artificial neurons*. These were originally conceived as simplified models of *biological* neurons, excitable nerve cells in the brain, which receive and transmit electrical signals. Such cells consist of multiple dendrites, which carry signals towards the cell body, and one single axon that carries nerve signals away from it. The axon branches into a number of axon terminals, which transmit signals via synapses to the dendrites of other nerve cells. See a drawing in Fig. 5.1 on the left.

Artificial neurons model this behavior by taking a number of inputs $\vec{x}$, *e.g.*, from other neurons, for which they produce a single output y. The neuron assigns weights to each input, modeling the synaptic strength between itself and previous neurons. These weights $\vec{w}$ are "learnable", and can be positive (excitatory) or negative (inhibitory). Initial research modeled neurons to always have a binary output, *e.g.* 0 or 1. The neuron sums all weighted inputs, and "fires" if that sum exceeds a certain threshold t:

$$y = \begin{cases} 1 & \text{if } \vec{x} \cdot \vec{w} > t, \\ 0 & \text{else.} \end{cases} \tag{5.1}$$

Artificial neurons were later modified to produce a continuous output,

$$y = f(\vec{x} \cdot \vec{w} + b), \tag{5.2}$$

with the addition of a learnable bias b, and the inclusion of a non-linear *activation function* f. A common choice for this function is the rectified linear unit, $\mathrm{ReLU}(z) = \max(0, z)$. Notice that both these equations are equivalent if $b = -t$ and f is a step function. Fig. 5.1 shows a schematic of the artificial neuron, and how it relates to its biological counterpart.

These neurons can be arranged to form *fully connected* or *dense* layers. Such a layer takes m inputs $\vec{x}$, and returns n outputs $\vec{y}$, corresponding to the number of neurons. Each neuron is connected to all inputs, hence the name. Denoting the layer's weights $W_{m\times n}$ and the biases $\vec{b}$, this can be formalized as,

$$\vec{y} = f(\vec{x}\, W_{m\times n} + \vec{b}). \tag{5.3}$$

The term $\vec{x}\, W_{m\times n} + \vec{b}$ is conventionally called a *linear projection* from the input dimension m to the output dimension n, and it will be notated as $\mathrm{Linear}_{m\to n}(\vec{x})$ in this thesis. The whole layer performs a non-linear projection then.

Stacking dense layers yields a fully connected or a dense neural network, which consists of one input layer, a number of hidden layers, and an output layer. Each layer's output corresponds to the next layer's input. See a schematic in Fig. 5.2. Stacked layers illustrate the importance of *non-linear* activation functions, since without them, multiple layers would be equivalent to a single layer. Dense networks are commonly called multi-layer perceptrons (MLPs) or feed forward networks (FFNs).

Having presented the basic form of a neural network, the question remains how its parameters should be chosen for the network to perform some specific task on a dataset. For a given dataset *sample* x, the network makes a *prediction* $\hat{y}$, which ideally corresponds to the true *target* value y. In the context of GW searches, the input may be detector data, and the network predicts 1 if that data contains a GW signal and 0 otherwise.

The process of optimizing a network's parameters is called *training*. In particular, this thesis is concerned with *supervised* learning, where this dataset is labeled, *i.e.* the true values are known. In this case, the dataset is split into training, validation and testing subsets. The

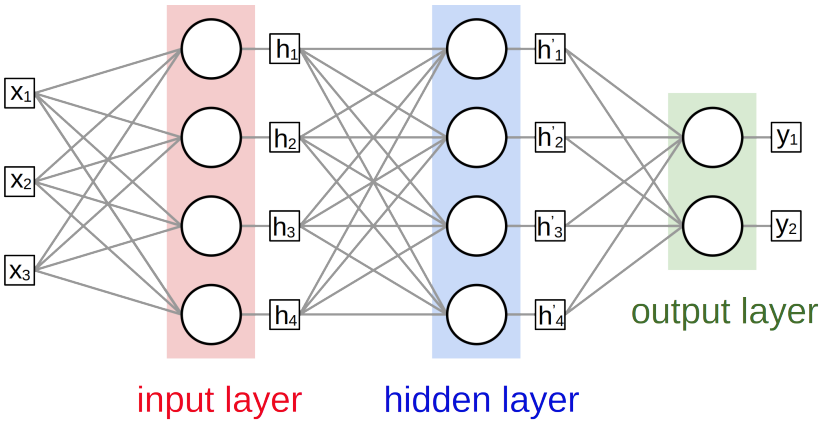


Figure 5.2: A dense neural network with three layers. Each dot denotes an artificial neuron. The network takes an input $\vec{x}$ and projects it to $\vec{h}$, and then $\vec{h}'$, of hidden dimension four. The final output layer projects $\vec{h}'$ to $\vec{y}$.

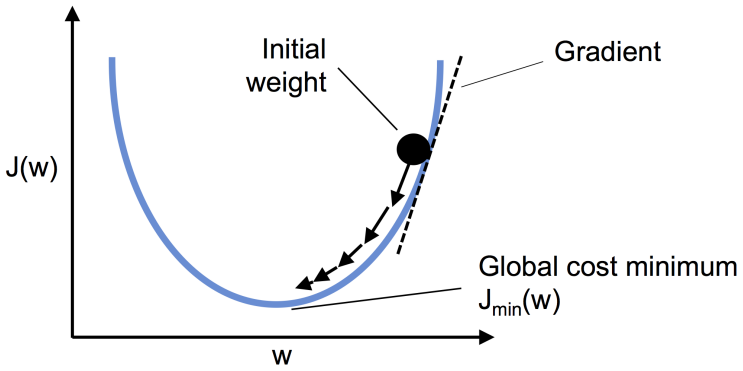


Figure 5.3: Gradient descent minimization of the loss J with respect to the network parameters or weights w. This is a simplified sketch, where w is one-dimensional. Contrast this with modern large language models with billions of parameters. Taken from [34].

parameters are optimized by minimizing a *loss function* $\mathcal{L}$ over the training set, *e.g.*, the mean-squared loss,

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}(x_i, \vec{\theta}) - y_i|^2, \tag{5.4}$$

where N is the number of training samples. Notice that the total loss depends solely on the network parameters $\vec{\theta}$. Formally, the optimal parameters are given by,

$$\vec{\theta}_{\text{opt}} = \arg\min_{\vec{\theta}} \mathcal{L}(\vec{\theta}). \tag{5.5}$$

Determining the minimum of $\mathcal{L}$ analytically is not possible once the model reaches a certain size. Instead, the parameters are initialized randomly and iteratively updated by calculating the gradient:

$$\vec{\theta}' = \vec{\theta} - \eta \, \vec{\nabla} \mathcal{L}(\vec{\theta}), \tag{5.6}$$

where η denotes the *learning rate*. This method is called *gradient descent*. The parameters follow the slope of the loss hypersurface towards its minimum (see Fig. 5.3). The loss gradient can be efficiently computed with *backpropagation*, which is simply based on the chain rule.

In practice, it is often not feasible or even necessary to update the parameters based on the loss of the whole training dataset. Instead, small batches are chosen randomly, for which the parameters are updated subsequently. Once the whole dataset is exhausted, the process is repeated. One full iteration over the training dataset is called an *epoch*.

5.2 Convolutional Neural Networks

CNNs were first developed for image analysis and were inspired by the organization of neurons in the brain's visual cortex [35]. In the visual cortex, neurons have a "small" receptive field, that is, they respond only to local regions of the visual field. The receptive fields of different neurons can overlap and together they cover the whole visual field. Moreover, neurons can specialize to recognize different local patterns, such as horizontal and vertical edges.

CNNs model this behavior by convolving learnable filters, called *kernels*, across the input. In practice, this operation is usually implemented as the cross-correlation between a kernel k and an input x. In one dimension, this is given by

$$(k \star x)(\tau) = \int_{\mathbb{R}} \mathrm{d}t \, k(t) \, x(t + \tau), \tag{5.7}$$

and yields a measure of similarity as a function of τ, the displacement of the input relative to the kernel. In discrete form, the convolution of a kernel $\vec{k}$ of size N over an input $\vec{x}$ outputs a *feature map* $\vec{y}$ with entries

$$y_i = (\vec{k} \star \vec{x})_i = \sum_{j=0}^{N-1} k_j \, x_{i+j}. \tag{5.8}$$

See Fig. 5.4, where two different kernels are applied to the same input image of a pagoda. One kernel extracts the input's vertical features, while the other extracts horizontal features. The convolution operation gives neural networks the ability to extract features in a spatially invariant manner.

A full CNN architecture will consist of multiple stacked *convolutional layers*. Each layer learns a number of kernels producing multiple feature maps, or *channels*. One layer's output

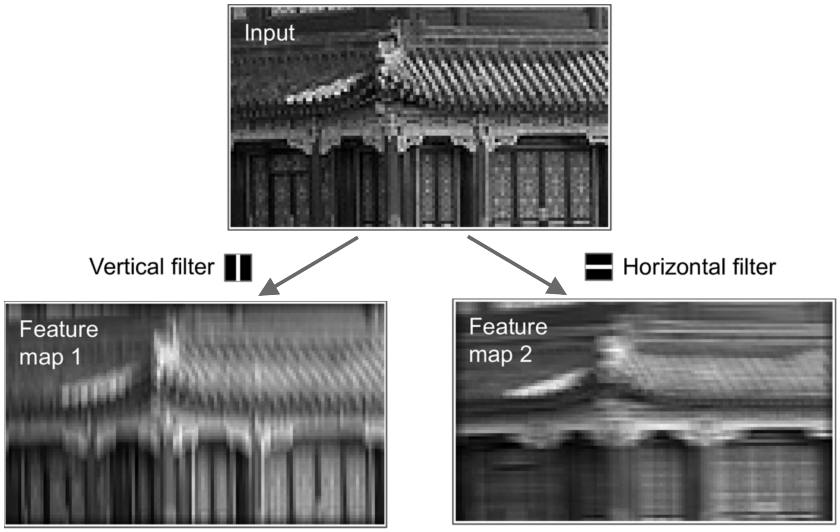


Figure 5.4: A "vertical" and a "horizontal" filter convolved over the image of a pagoda. In the feature map on the left, the vertical features stand out more. *e.g.* the roof tiles and the columns. On the right they appear less bright and more blurry. Adapted from [35].

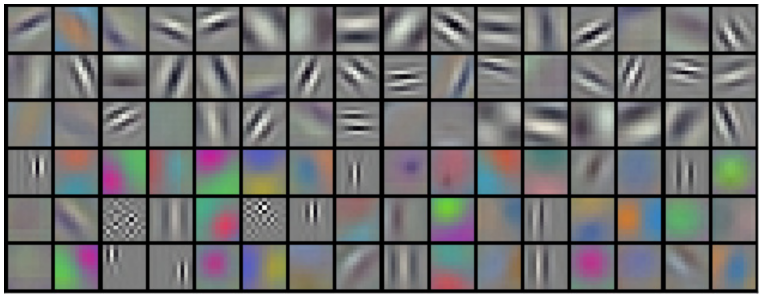


Figure 5.5: The initial convolutional layer's 96 kernels in AlexNet [36]. These are of shape $3 \times 11 \times 11$, and plotted as RGB images.

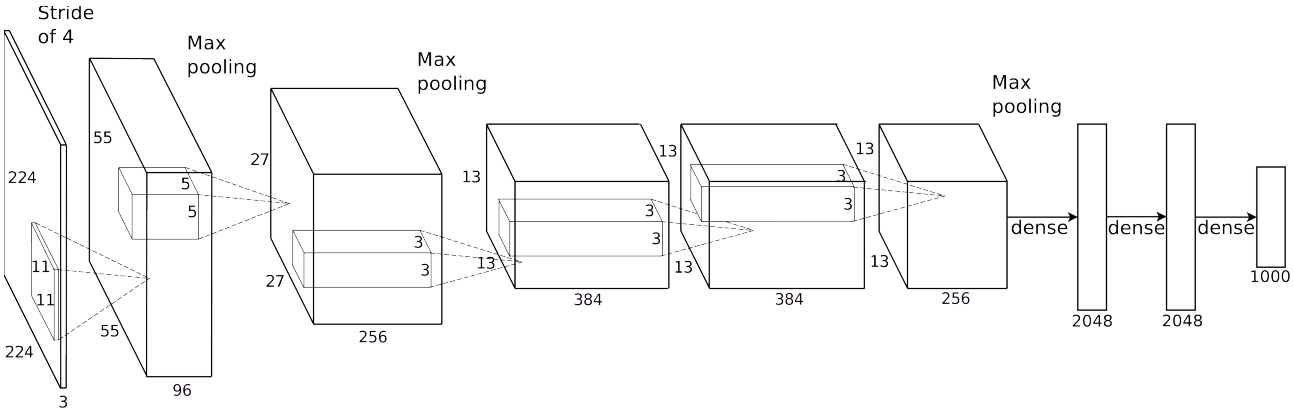


Figure 5.6: AlexNet architecture [36]. Multiple stacked convolutional layers with ReLU and pooling in between, and a final dense network. The schematic has been adapted.

gets fed to the next layer. This way, the complexity of the features that are extracted from the input gradually increases. The initial layers will learn low-level features, *e.g.* vertical or horizontal edges, while the later layers can extract high-level features, *e.g.* human faces or GW signals. See the learned low-level kernels of a 2D-CNN, AlexNet [36], in Fig. 5.5.

Generally, a 1D convolutional layer takes an input tensor X of shape $C_{\text{in}} \times L$ and produces an output of shape $C_{\text{out}} \times L$. The sequence length is L, and C_{in} and C_{out} denote the number of input and output channels. The output of such a layer is then,

$$\text{ConvLayer}_{C_{\text{in}} \to C_{\text{out}}}(X) = \operatorname*{concat}_{i=0}^{C_{\text{out}}-1} \sum_{j=0}^{C_{\text{in}}-1} \vec{k}_{i,j} \star \vec{x}_j, \tag{5.9}$$

with $\vec{k}_{i,j}$, the kernel corresponding to the i-th output channel and the j-th input channel.

Usually, the number of channels increases the deeper the CNN gets. To counteract the increased computational burden this introduces, CNNs often perform downsampling on the feature maps between each layer. This can be achieved by *striding*, which disregards some of the feature map entries in an orderly fashion, *e.g.* every second, which corresponds to a stride of two. Another way of downsampling is *pooling*, where the feature map is subdivided into equal sized regions, and for each region only the maximum value or the mean value is extracted.

Most CNNs apply a number of convolutional layers, each followed by a ReLU non-linearity and downsampling. Without this non-linearity, multiple convolutional layers would be equivalent to a single layer, similar to stacked fully-connected layers. The final feature maps are usually flattened and fed to a fully-connected network to return an output of the desired size. See an example 2D-CNN in Fig. 5.6.

In GW signal detection, both two and one-dimensional CNNs are employed. In the former case, the network's input can be the strain spectrogram [8], and in the latter case, it is the strain time series [9]. Notice especially, the similarity of 1D-CNNs with matched filtering, which is based on "de-noising" the detector strain $s(t)$ with a filter function $K(t)$ via $\int \mathrm{d}t K(t)s(t)$ (see Eq. 4.8). In a way, the whole set of a CNN's kernel parameters can be thought of as a particularly efficient way of storing waveform templates.

5.3 Transformers

The Transformer [11] was originally developed for natural language processing tasks (NLP), in particular for machine translation. Unlike CNNs, which rely on local receptive fields, Transformers model global dependencies through *attention mechanisms*. Transformers generalize well to other sequential data tasks, *e.g.* image or audio related. A transformer-based model has proved successful in GW parameter estimation under ET conditions with signal overlap [10]. This motivates the investigation of transformers for GW searches.

Attention Mechanism

The central component of the Transformer architecture is the attention mechanism, which builds on [37]. The mechanism takes two sequences as input. Each sequence comprises a number of tokens, which, *e.g.* in NLP, can correspond to the words of a sentence. The two inputs are referred to as the *source* sequence and the *target* sequence. If both are the same, this is called *self-attention*, and if they are different, it is *cross-attention*. The attention mechanism

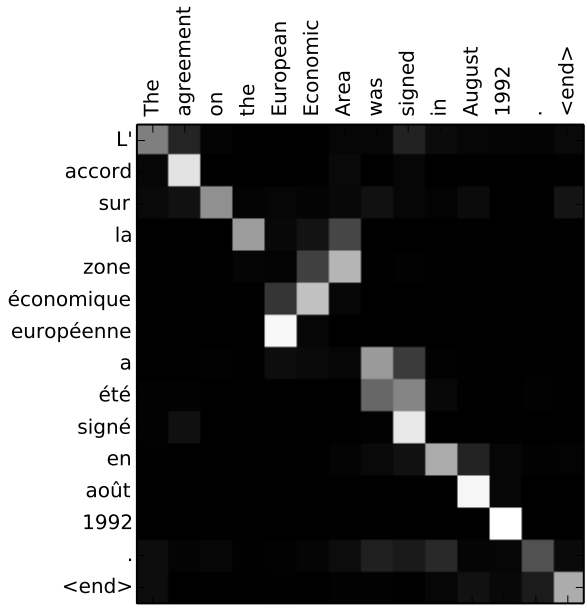


Figure 5.7: Cross-attention matrix in an English-to-French translation model [37]. The English sentence is the source sequence, and the French translation the target sequence. Lighter colors mean higher values.

relies on *attention weights* to model the relationship between source and target sequence. These weights correspond to a $N_{\text{tgt}} \times N_{\text{src}}$ matrix. See an example in Fig. 5.7, the attention matrix in an English-to-French translation model, capturing the relationship between "area" in English and "zone" in French despite their different ordering in the two languages.

Scaled Dot-Product Attention

In transformers, the attention mechanism is inspired by a search algorithm for a database of *key-value* pairs. The keys serve as identifiers and the values contain the actual data associated with them. The database is efficiently searched by matching a *query* against the keys to retrieve the relevant values. For each query, the attention mechanism returns a weighted sum of all the values. The weights correspond to the similarity between the query and all keys, which is computed via their dot product.

This is implemented by linearly projecting the keys and values from the source sequence tokens, and the queries from the target sequence. The input tensors are denoted as $S_{N_{\text{src}} \times d_{\text{src}}}$ and $T_{N_{\text{tgt}} \times d_{\text{tgt}}}$, meaning source and target sequence respectively. Each sequence consists of a number of tokens N, which are vectors of size d. The queries, keys and values are then,

$$Q_{N_{\text{tgt}} \times d_{\text{embed}}} = \text{Linear}_{d_{\text{tgt}} \to d_{\text{embed}}}(T) \tag{5.10}$$

$$K_{N_{\text{src}} \times d_{\text{embed}}} = \text{Linear}_{d_{\text{src}} \to d_{\text{embed}}}(S) \tag{5.11}$$

$$V_{N_{\text{src}} \times d_{\text{embed}}} = \text{Linear}_{d_{\text{src}} \to d_{\text{embed}}}(S), \tag{5.12}$$

respectively, with the embedding dimension d_{embed}. (Notated like this, the linear projection only applies to the final tensor dimension, and its weights are shared across all other dimensions.)

From these, for each query vector $\vec{q}_i$, its weights $\vec{w}_i$ are computed as the dot product between that query vector and all keys:

$$\vec{w}_i = \text{softmax}\begin{pmatrix} \vec{q}_i \cdot \vec{k}_0/\sqrt{d_{\text{embed}}} \\ \vec{q}_i \cdot \vec{k}_1/\sqrt{d_{\text{embed}}} \\ \vdots \\ \vec{q}_i \cdot \vec{k}_{N_{\text{src}}-1}/\sqrt{d_{\text{embed}}} \end{pmatrix}. \tag{5.13}$$

The dot product is scaled by the embedding dimension. *Softmax* is applied to ensure that the weight vector sums to one. For an arbitrary vector $\vec{x}$ of size N the softmax operation returns

$$\text{softmax}\,\vec{x} = \frac{1}{\sum_i \mathrm{e}^{x_i}} \begin{pmatrix} \mathrm{e}^{x_0} & \mathrm{e}^{x_1} & \dots & \mathrm{e}^{x_{N-1}} \end{pmatrix}^\intercal. \tag{5.14}$$

The whole attention matrix has these weight vectors as its rows, *i.e.*,

$$W_{N_{\text{tgt}} \times N_{\text{src}}} = \begin{pmatrix} \text{——} & \vec{w}_0^\intercal & \text{——} \\ \text{——} & \vec{w}_1^\intercal & \text{——} \\ & \vdots & \\ \text{——} & \vec{w}_{N_{\text{tgt}}-1}^\intercal & \text{——} \end{pmatrix}. \tag{5.15}$$

Computing the attention weights as such is referred to as *scaled dot-product attention*.

The attention mechanism will return a tensor of size $N_{\text{tgt}} \times d_{\text{embed}}$. To this end, the weight matrix and value tensor are multiplied. This yields a tensor, where each row is a weighted sum, $\sum_j w_{ij}\vec{v}_j$, corresponding to the i-th query. A final output linear projection is applied. The whole attention mechanism can be stated as,

$$\text{Attention}(Q, K, V) = \text{Linear}_{d_{\text{embed}} \to d_{\text{embed}}}\left(\text{softmax}\left(\frac{QK^\intercal}{\sqrt{d_{\text{embed}}}}\right)V\right), \tag{5.16}$$

with the attention matrix W somewhat confusingly denoted as softmax $(QK^\intercal/\sqrt{d_{\text{embed}}})$, where softmax is applied along the final dimension.

Considering English-to-French machine-translation, the English sentence is the source sequence, while the target sequence somehow represents the French translation. (The correct translation is obviously not known beforehand. How the Transformer gains this representation will be covered later.) The projected values represent the English words in some abstract way, and the keys serve as labels for these values. Similarly, the query tokens are projected from the target sequence and encode what information is needed at each position. The French word corresponding to one of the queries is obtained by a weighted sum of all value tokens. The network learns to assign larger weights to the values most relevant to that position. A final linear layer projects this weighted sum of abstract English word representations into a French word representation.

Multi-Head Attention

Transformers develop the attention mechanism further to include multiple *heads*. Each head learns its own query, key and value projections, and performs the attention mechanism independently of other heads. The idea is that each head can learn to perform different tasks (see Fig. 5.8). This is called *multi-head attention* (MHA).

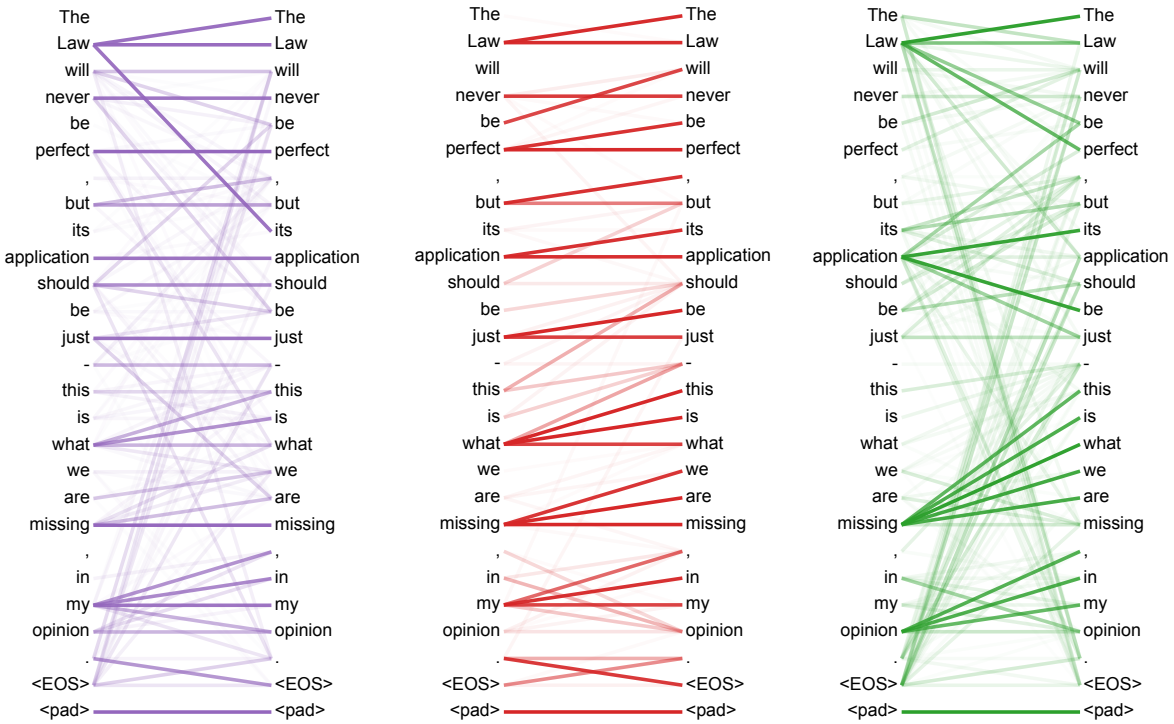


Figure 5.8: Self-attention weights of different heads in the original Transformer [11].

In practice, to not introduce additional computational strain, the embedding dimension inside each head is reduced. This is usually implemented by keeping the original projections (Eqs. 5.10, 5.11, 5.12), and splitting each tensor into N_{heads} heads along the embedding dimension to yield tensors Q_i of size $N_{\text{tgt}} \times d_{\text{embed}}/N_{\text{heads}}$ and K_i, V_i of shape $N_{\text{src}} \times d_{\text{embed}}/N_{\text{heads}}$. For this to work, d_{embed} has to be divisible by the total number of heads. The output of all heads is concatenated along their final dimension, and a linear projection is applied:

$$\text{MHA}(Q,K,V) = \text{Linear}_{d_{\text{embed}} \to d_{\text{embed}}}(\underset{i=0}{\overset{N_{\text{heads}}-1}{\text{concat}}}(\text{softmax}\left(\frac{Q_i K_i^\intercal}{\sqrt{d_{\text{embed}}/N_{\text{heads}}}}\right)V_i)). \tag{5.17}$$

Notice that for a single head, this is equivalent to Eq. 5.16.

Model Architecture

The Transformer follows an encoder–decoder architecture (see Fig. 5.9). The encoder processes the input sequence into an abstract representation, and the decoder generates the output sequence from this representation. The decoder predicts the output sequence in an autoregressive manner, that is, token by token, starting with a beginning-of-sentence token. This makes sense for machine translation, while for other tasks the decoder can also be adapted to make predictions in parallel instead. Modern transformer models often employ only the encoder or decoder.

Both the encoder and decoder are composed of identical stacked layers, where each layer's output is the next one's input. These layers comprise sub-layers of multi-head attention and dense networks, *i.e.* feed-forward networks (FFNs).

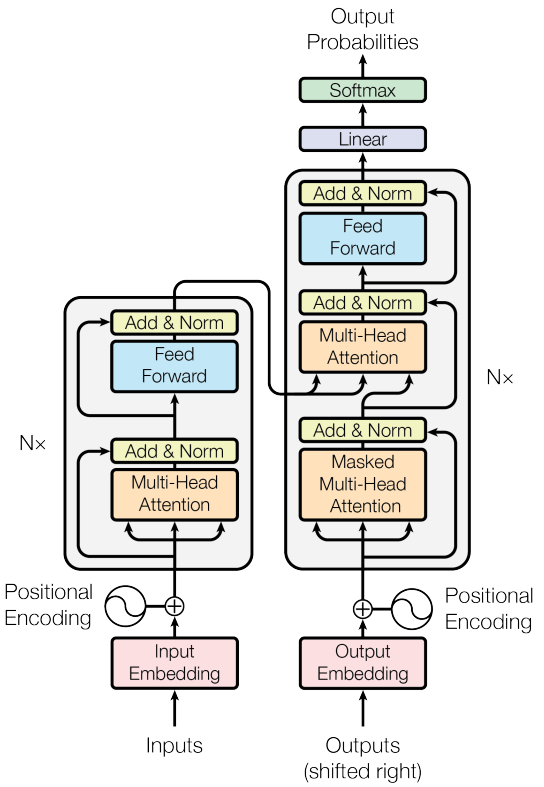


Figure 5.9: The original Transformer architecture [11].

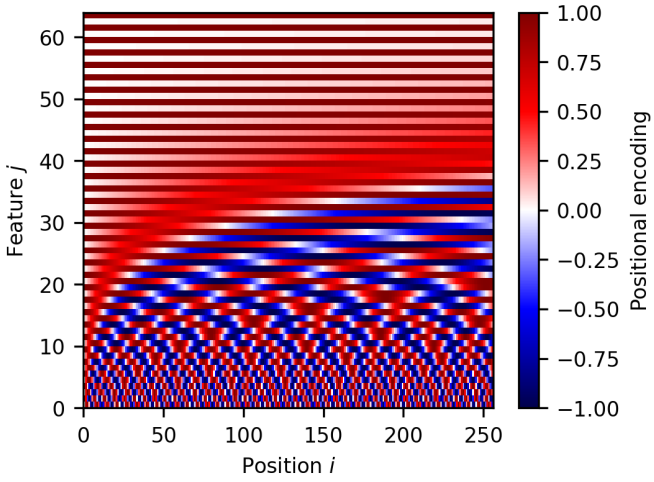


Figure 5.10: Sinusoidal positional encodings.

The encoder layers apply two sub-layers, self-attention followed by an FFN. The decoder layers apply three sub-layers instead: *masked* self-attention, followed by cross-attention and an FFN. Masking the self-attention sub-layer ensures that the decoder can only attend to previously predicted tokens. Transformer models, which predict tokens in parallel, omit this masking. The cross-attention's source sequence is the encoder output and the target sequence is obtained from the previous sub-layer.

The architecture makes use of *residual connections* for all sub-layers, where the input is added to the output, *i.e.*, $x + \text{SubLayer}(x)$. This works, since both these sub-layers return a tensor of the same dimension as the input. The FFN is,

$$\text{FFN}(x) = \text{Dropout}(\text{Linear}_{d_{\text{ff}} \to d_{\text{embed}}}(\text{GELU}(\text{Linear}_{d_{\text{embed}} \to d_{\text{ff}}}(x)))), \tag{5.18}$$

with hidden dimension d_{ff}, and a Gaussian error linear unit (GELU) activation [38]. Dropout [39] is a regularization technique, which prevents networks from overfitting by randomly zeroing out elements of the given tensor. Dropout is not applied during inference. In the original architecture, the residual connection is followed by a *layer normalization* [40] to standardize the output along the embedding dimension. This is called *post*-layer normalization. To ease training, modern implementations often employ *pre*-layer normalization, so $x + \text{SubLayer}(\text{LayerNorm}(x))$ instead.

Unlike in CNNs, the sequential nature of the data is not baked into the Transformer architecture itself. Thus, *positional encodings* are added to encoder and decoder input tokens. The original Transformer makes use of so-called fixed sinusoidal encodings. For a token

position $i < N_{\text{tokens}}$ and a feature $j < d_{\text{embed}}$, this is:

$$P(i, j) = \begin{cases} \sin\left(i/N^{\frac{j}{d_{\text{embed}}}}\right) & \text{if } j \text{ even,} \\ \cos\left(i/N^{\frac{j-1}{d_{\text{embed}}}}\right) & \text{if } j \text{ odd,} \end{cases} \tag{5.19}$$

where $N = 10^4$. See the encodings of 256 tokens with an embedding dimension of 64 plotted in Fig. 5.10.

6 Data Simulation

Since the Einstein Telescope is currently still in its planning stages, no real detector data exist, with which neural networks can be trained and tested. Thus, a considerable amount of this thesis' work has been spent on simulating gravitational wave signals, and the corresponding detector response under realistic noise conditions. In the context of machine learning, such simulated data is commonly referred to as *synthetic* data.

This chapter provides a detailed description of the simulation of detector noise and GW signals, as well as the construction of samples to form datasets. Most of the utilities required for these simulations are provided by the PyCBC software package [30], and the hardware was provided by the PALMA II HPC of the University of Münster. The whole simulation script is available at [41].

6.1 Detector Noise

This thesis assumes the detector noise to be *Gaussian*, meaning it has amplitudes, which are normally distributed. The noise is then fully characterized by the noise power spectral density (PSD). See Ch. 4 for further information. This thesis employs the sensitivity of ET according to the latest design report [23] in a triangular configuration with 10 km-long arms (see Ch. 3). The corresponding PSD data are taken from [42], which provides the noise PSDs for the LF and HF interferometers in the frequency range from 1 Hz to 10 kHz, as well as for their combined sensitivity (LF+HF). The combined PSD is obtained through,

$$\frac{1}{S_n^{\mathrm{LF+HF}}(f)} = \frac{1}{S_n^{\mathrm{LF}}(f)} + \frac{1}{S_n^{\mathrm{HF}}(f)}. \tag{6.1}$$

The corresponding PSD curves are shown in Fig. 6.1. In this work, the intricacies of the xylophone configuration, where each detector consists of two nested interferometers, are ignored. Instead, ET is treated as three interferometers in a triangular configuration, assuming the ET LF+HF noise PSD for each.

Such noise is called *colored*, because its power spectrum is *not* constant over the whole frequency regime (in contrast to *white* noise with a flat power spectrum), and *stationary* because the power spectrum stays constant with respect to time. Notice that for currently operating GW interferometers such as LIGO and Virgo their noise is typically non-stationary. This is due to slow drifts in the PSD over minutes or hours and because of short-duration transient noise events called *glitches* [43]. Regardless, in this work the detectors' noise is assumed to be stationary, since it is not entirely clear if and how exactly non-stationarity might arise.

In the proposed ET xylophone configuration, some of the interferometers' test masses will be located less than a few 100 m apart, which is why a non-negligible correlation is expected among the noise of the detectors in the form of seismic noise, especially in the low-frequency

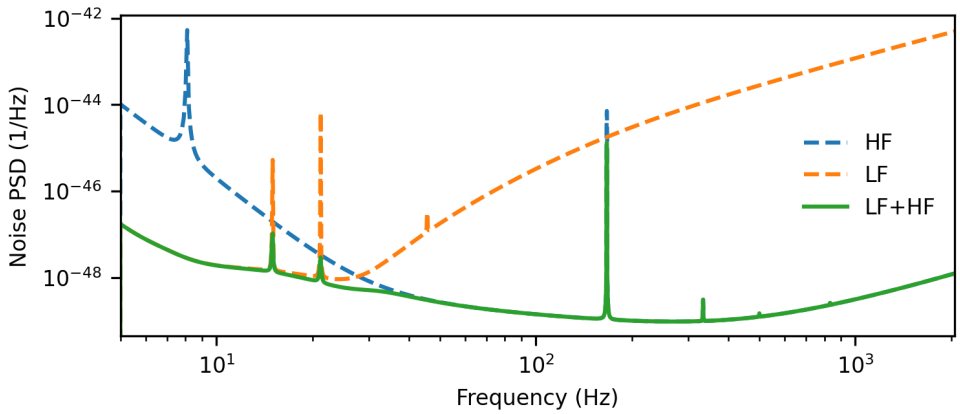


Figure 6.1: The ET noise PSD of the LF and HF interferometers, and of their combination (LF+HF), which is the PSD employed in this work. The PSDs are plotted from 4 Hz to 2048 Hz, corresponding to the simulation upper and lower frequency range. Data taken from [42].

regime up to ~40 Hz [44]. Currently, there is no routine implemented in PyCBC for the simulation of correlated noise. Instead, this is done from scratch following the approach in [45]. The algorithm will be presented by first describing how uncorrelated noise is simulated from a given power spectrum, and then showing how this procedure is adapted in the case of correlated noise.

Uncorrelated Noise

Consider a noise time-series $n = (n(t_0), n(t_1), \dots)$ that is to be simulated for one detector with a PSD $S_n = (S_n(f_0), S_n(f_1), \dots)$. The discrete Fourier transform of that time-series $\tilde{n} = (\tilde{n}(f_0), \tilde{n}(f_1), \dots)$ should then satisfy,

$$\frac{1}{2} S_n(f) = \Delta f \left\langle |\tilde{n}(f)|^2 \right\rangle, \tag{6.2}$$

where Δf is the frequency resolution. This follows from the PSD definition (Eq. 4.2), as evaluated over a constant observation time T, in discrete form. Requiring the noise to be Gaussian, motivates an ansatz of the form,

$$\tilde{n}(f) = a(x + iy), \tag{6.3}$$

where x and y are drawn from a normal distribution with zero mean and unit variance, $x, y \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$, and where a is some arbitrary prefactor. Since the expectation value of a squared normal random variate is one, $\left\langle |\tilde{n}(f)|^2 \right\rangle = a^2(\langle x^2 \rangle + \langle y^2 \rangle)$ simplifies to $2a^2$, and thus with Eq. 6.2,

$$a = \frac{1}{2} \sqrt{\frac{S_n(f)}{\Delta f}}. \tag{6.4}$$

That is, the discrete Fourier transform of the noise has a uniformly random phase and a magnitude, which is, on average, proportional to the square root of the noise PSD. Finally, n can be computed as the inverse discrete Fourier transform of $\tilde{n}$.

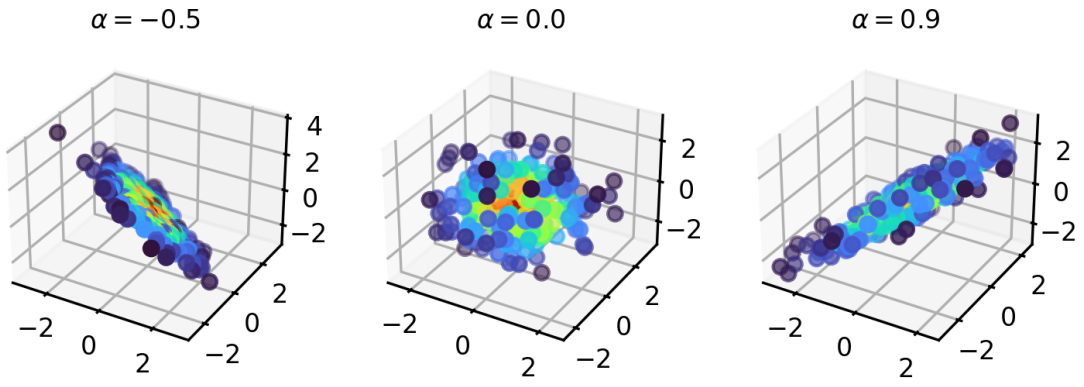


Figure 6.2: Points drawn from a centered three-variate normal distribution with unit variance in the case of negative correlation (*left*), zero correlation (*center*) and positive correlation (*right*). The points' color signify the probability density at that location.

Correlated Noise

For correlated noise, the problem at hand is similar: For multiple detectors, noise time-series n_i should be simulated from a *spectral matrix* $\underline{\underline{S_n}}$. This matrix contains each detector's noise PSD on its diagonal and the *cross spectral densities* (CSDs) between detectors on the off-diagonal entries. For ET, that means three detectors, assuming the same PSD as in the uncorrelated case is each detector, $S_n^{ii} = S_n$ for $i = 1, 2, 3$. The CSDs are given by $S_n^{ij} = \alpha_{ij}\sqrt{S_n^{ii} S_n^{jj}} = \alpha_{ij} S_n$ with some correlation coefficient α_{ij}. For the sake of simplicity, the same correlation is assumed between all detectors. Then, the spectral matrix can be written as

$$\underline{\underline{S_n}} = \begin{pmatrix} S_n^{11} & S_n^{12} & S_n^{13} \\ S_n^{21} & S_n^{22} & S_n^{23} \\ S_n^{31} & S_n^{32} & S_n^{33} \end{pmatrix} = \begin{pmatrix} 1 & \alpha & \alpha \\ \alpha & 1 & \alpha \\ \alpha & \alpha & 1 \end{pmatrix} S_n, \tag{6.5}$$

with α bounded to a range between -0.5 and 1. For a proof, see the [45].

An ansatz similar to Eq. 6.3 is made for each detector's noise Fourier transform,

$$\tilde{n}_i(f) = a_i(x_i + iy_i). \tag{6.6}$$

But now, the random variates x_i and y_i for each detector are not drawn independently of each other. Rather, vectors $\vec{x} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^\mathsf{T}$ and $\vec{y} = \begin{pmatrix} y_1 & y_2 & y_3 \end{pmatrix}^\mathsf{T}$ are drawn from a *multivariate* (in this case three-variate) normal distribution again with zero mean and unit variance, *i.e.*, with a *covariance matrix* that is given by

$$\Sigma = \begin{pmatrix} 1 & \alpha & \alpha \\ \alpha & 1 & \alpha \\ \alpha & \alpha & 1 \end{pmatrix}. \tag{6.7}$$

For a visualization of this distribution, see Fig. 6.2. Again, for each detector Eq. 6.2 should hold. Since $\langle x_i^2 \rangle = 1$, the same formula for a_i is recovered as in Eq. 6.4. Then, the inverse Fourier transform yields the desired time-series.

In practice, a small positive correlation of $\alpha = 0.1$ is assumed in the regime where seismic and Newtonian noise is dominant, which is below frequencies of 10 Hz. For higher frequencies $\alpha = 0$, in which case the covariance matrix collapses to an identity matrix and the probability

distribution effectively behaves as three independent normal distributions. A lower frequency cutoff of 4 Hz is applied, which means that the PSD is set to zero below that frequency. Furthermore, a sampling rate of 2048 Hz is chosen resulting in an upper resolution limit given by the Nyquist frequency, in this case 1024 Hz. This is a reasonable resolution range, when considering BBH signals only. The inclusion of BNS signals in the simulation as well would necessitate a higher sampling rate.

6.2 Gravitational Wave Signals

It is important for the simulated dataset to cover all possible signal waveforms evenly such that when training neural networks no biases are learned. The shape of these waveforms will depend on a number of parameters, which are referred to as extrinsic and intrinsic. The former describe the signal source's location with respect to the detector reference frame and the latter are the intrinsic properties of the binary system, such as their masses.

Parameters

The extrinsic parameters are defined through three simulation frames, the detector frame, the radiation frame and the source frame. In this thesis, the detector frame is the Equatorial coordinate system. This frame has a z-axis pointing towards the Earth's north pole, and an x-axis that points towards the March equinox. The y-axis is determined by the right circular convention. The radiation frame can be thought of as the sky projection of the detector frame. Its z-axis is the line-of-sight vector towards the GW source. The x-axis is the projection of the detector frame's x-axis onto the sky plane, and the y-axis is again known by convention. The source frame has a z-axis pointing along the binary's orbital momentum vector. It x–y plane is the orbital plane, where the x-axis the projection of the line-of-sight vector. See a schematic of the simulation frames in Fig. 6.3.

There are a total of seven extrinsic parameters. The polar coordinates of the line-of-sight vector, *i.e.* the inverse GW propagation direction, are the declination θ and the right ascension ϕ. These two angles are defined with respect to the detector frame. The orientation of the binary with respect to the radiation frame yields two more parameters, the inclination ι and the polarization ψ. These are the polar angles of the source frame's z-axis, *i.e.* the binary plane's normal vector, defined within the radiation frame. The other three parameters are the distance between detector and source d, as well as the coalescence time t_{coa} and corresponding coalescence phase ϕ_{coa}. The latter is the polar angle of primary mass in the source frame. The definition of the angle parameters are shown in Fig. 6.4.

The binary system has six intrinsic parameters. These include the two masses, the larger of which is called the primary mass m_1 by convention. (The other one is denoted m_2.) These masses can have non-zeros spins, which are described by the two vectors $\vec{S}_1$ and $\vec{S}_2$. The spins are defined within the source frame. In practice it is useful to define the dimensionless spin as $\vec{\chi}_i = \vec{S}_i/m_i^2$ with $|\vec{\chi}_i| < 1$ (in natural units that is).

Waveform Simulation

For the simulation of gravitational wave signals from the inspiral, merger and ringdown of black hole binaries, the phenomenological model *IMRPhenomD* [47] is used as implemented by *lalsimulation*. This is part of the LIGO Algorithm Library [48], which in turn is accessed via PyCBC.

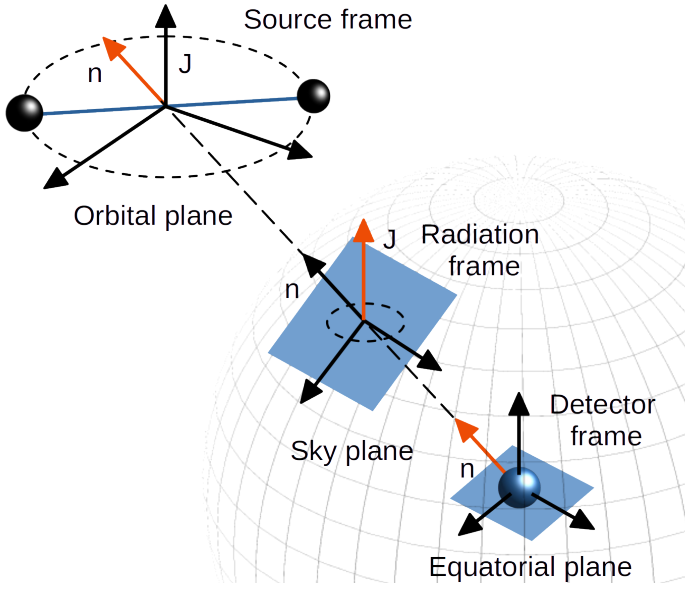


Figure 6.3: The three simulation frames. $\vec{n}$ is the line-of-sight vector that points towards the source. $\vec{J}$ denotes the binary's angular momentum vector.

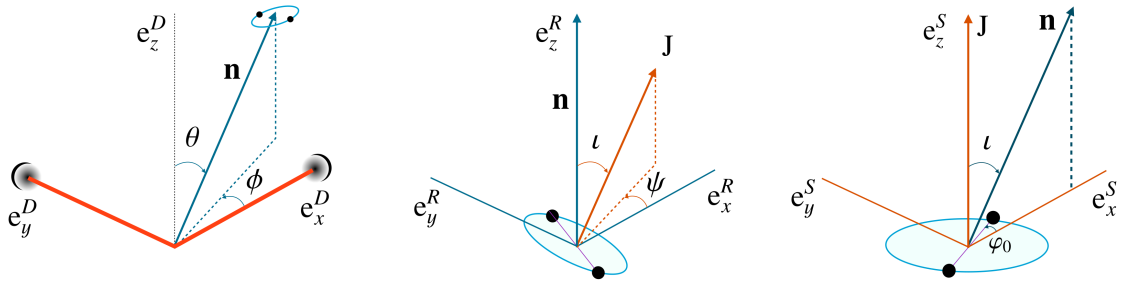


Figure 6.4: The extrinsic parameters as defined within the three simulation frame. Here, the detector frame is defined by the interferometer geometry, which is not the convention this thesis follows. Taken from [46].

These simulations take all the intrinsic parameters as inputs, which are the masses and spins. The black hole mass distribution is modeled after CBC detections made by the second-generation network so far (see Fig. 3.3). Both black hole masses sampled uniformly in $[10\,\mathrm{M}_\odot, 80\,\mathrm{M}_\odot]$, where the larger is the primary mass m_1 and the other m_2. This roughly corresponds to coalescence frequencies in a range of $[30\,\mathrm{Hz}, 220\,\mathrm{Hz}]$ (see Eq. 2.23), making the chosen sampling rate of 2048 Hz appropriate. In the scope of this work, only aligned spin BBHs are considered, which means $\chi_x^i = \chi_y^i = 0$ and $\chi_z^i \geq 0$. The z-component of the spin is sampled uniformly in $[0, 0.998]$ with an upper limit that equals the Thorne limit.

The radiation pattern of the binary also depends on the binary's inclination and the coalescence phase. The inclination ι is sampled from a sine distribution with a probability density function proportional to $\sin x$ for x in $[0, \pi]$. The coalescence phase φ_{coa} is sampled uniformly in $[0, 2\pi)$. Another simulation input parameter is the source distance. Since only the total strain amplitude depends on that distance and since there will be a rescaling step later anyway, the exact value is arbitrarily set to 100 Mpc. The simulation employs a lower frequency cutoff, which is the starting frequency of the waveform, which is set to 4 Hz. A waveform with all these parameters sampled accordingly can be seen plotted in Fig. 6.5.

Antenna Projection

The simulated waveform are projected onto the ET interferometers to yield the GW strain according to Eq. 3.3. A PyCBC routine is relied upon to handle these antenna projections. The package implements ET according to [49] as three detectors in a triangular configuration with 10 km-long arms. The detector is located at 43.6314 °N 10.5045 °E, which is the site of Virgo.

PyCBC takes as input the detector GPS location, the GW signal sky position (in the equatorial frame) and the polarization. In the simulation, the sky location is sampled uniformly. This means, the declination θ is sampled from a sine distribution and the right ascension ϕ uniformly in $[0, 2\pi)$. The GW polarization ψ is sampled in $[0, 2\pi)$. The routine further depends on the GW arrival time, for which PyCBC calculates the revolution of the Earth and updates the detector GPS location accordingly. Since the GW sky position is sampled uniformly anyway, the arrival time is arbitrarily chosen to be zero. A GW signal that is projected as such is plotted in Fig. 6.6.

Waveform Tapering

The waveform simulation starts at the lower frequency cutoff, which is set to 4 Hz. This results in an "abrupt" jump in strain amplitude once the signal sets in which is an unphysical signature a neural network might pick up on easily. To mitigate this a tapering of the waveform is implemented by multiplying with a one-sided Tukey window:

$$w(x) = \begin{cases} \frac{1}{2}(1 - \cos\frac{2\pi}{\alpha}x) & \text{if } 0 \leq x < \frac{\alpha}{2} \\ 1 & \text{if } \frac{\alpha}{2} \leq x < 1. \end{cases} \tag{6.8}$$

Here, x is in units of total waveform length, *i.e.*, $x = 0$ corresponds to the beginning and $x = 1$ to the end of the waveform. $\alpha = 1/4$ is chosen which means that the smoothing extends up to one eighth of the waveform. This trick is borrowed from [9].

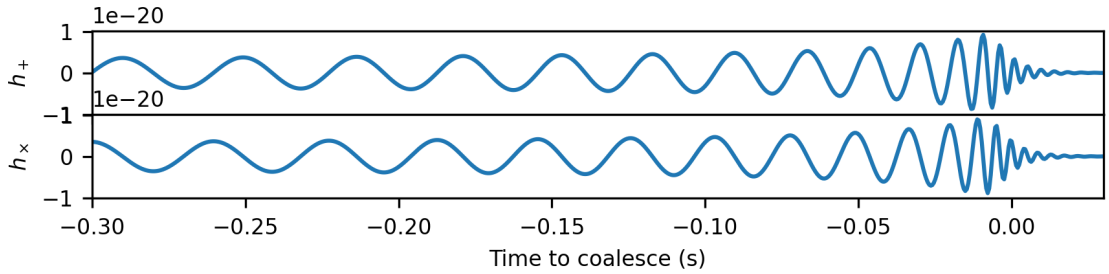


Figure 6.5: The plus and cross polarization strain of a GW signal emitted by a black-hole binary with masses $m_1 = 50\,\mathrm{M}_\odot$, $m_2 = 40\,\mathrm{M}_\odot$ and spins $\chi_z^1 = 0.73$, $\chi_z^2 = 0.43$ at a distance of 100 Mpc.

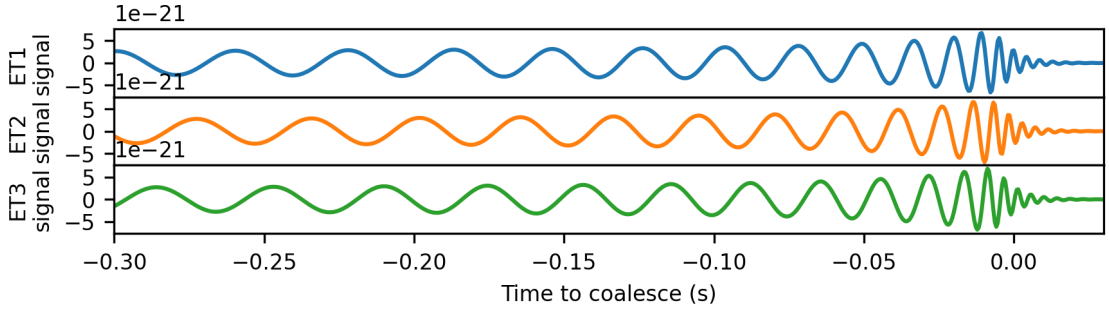


Figure 6.6: The same GW signal from Fig. 6.5 projected onto the three ET detectors as having a sky location $\phi = 263°$, $\theta = 56°$ and a polarization that is $\psi = 78°$.

SNR Rescaling

In order to ease training for the neural networks, the dataset should include signals with a "loudness" that is evenly distributed. Preferably in a range, where detection is neither impossible nor trivial. To achieve this, the waveforms are rescaled by some factor β. A measure for the signal loudness is the optimal SNR (see Sec. 4). This is denoted ρ instead of ρ_{opt} from now on. For multiple detectors the optimal SNR is,

$$\rho^2 = \rho_1^2 + \rho_2^2 + \rho_3^2, \tag{6.9}$$

with ρ_i^2 denoting the SNR of each of the three ET detectors. The individual detector SNR is given by Eq. 4.10, or in discrete form by,

$$\rho_i^2 = 4 \sum_j \Delta f \frac{|\tilde{h}_i(f_j)|^2}{S_n(f_j)}. \tag{6.10}$$

Rescaling the signal amplitude by some positive factor $h_i(t) \to \beta h_i(t)$, means that the Fourier transform scales as, $\tilde{h}_i(f) \to \beta \tilde{h}_i(f)$. Thus, the individual detector SNRs scale as, $\rho_i \to \beta \rho_i$ and finally the total SNR as, $\rho \to \beta \rho$. That is to say, the signal waveform can just be rescaled by some positive prefactor to ensure a desired SNR. This prefactor is then given by $\beta = \rho/\rho'$, where ρ is the desired SNR and ρ' the SNR prior to rescaling. This desired SNR is sampled uniformly in $[5, 15]$. Signals with an SNR of ~ 5 are more or less not detectable with matched filtering. For SNRs over ~ 10, matched filtering reaches close to perfect detection ratios [50]. GW150914 was observed with an total SNR of 24 [3].

In the case of correlated noise, this SNR formula does not hold anymore. Instead, as proposed in [45], the SNR is given by,

$$\rho^2 = 4\Delta f \sum_k \sum_{i,j} \tilde{h}_i(f_k)^* (\underline{\underline{S_n}}^{-1})_{ij}(f_k) \tilde{h}_j(f_k), \tag{6.11}$$

where i, j sum over all detector combinations and $\underline{\underline{S_n}}^{-1}$ is the inverse spectral matrix,

$$\underline{\underline{S_n}}^{-1} = \frac{1}{2\alpha^2 - \alpha - 1} \begin{pmatrix} -\alpha - 1 & \alpha & \alpha \\ \alpha & -\alpha - 1 & \alpha \\ \alpha & \alpha & -\alpha - 1 \end{pmatrix} \frac{1}{S_n}. \tag{6.12}$$

Fortunately, since ρ^2 is just a linear combination over all $\tilde{h}_i(f)^* \tilde{h}_j(f)$, rescaling with $\beta = \rho/\rho'$ is still valid. The difference is only that the pre-rescaling SNR ρ' is calculated using the adapted SNR formula, *i.e.* Eq. 6.11.

6.3 Samples and Datasets

The datasets are built from multiple samples. Each sample resembles ET detector stream data of some length. To this end, partially correlated noise of that length is simulated and a number of GW signals are *injected*, which simply means added. The waveform parameters are drawn randomly as described previously. For better clarity, the parameter distributions are restated in Tab. 6.1. The signals are injected with random coalescence times, covering the whole length of the detector stream.

In this work, the task of detecting multiple long-duration signals will be split in two steps. First, the simplified problem of detecting one long-duration signal per sample is considered. This allows testing different architectures, while increasing the sample length. To fit a full BBH waveform, a final sample length of ≈ 1 min is aimed for. Multiple datasets with increasing length are constructed. For better comparability, training datasets will always consist of 1 M samples in this thesis. Since there is only up to one signal per sample, this task can be simplified to a *binary-classification* problem. In this case, the network will predict one number per sample that signifies whether there is signal in the given detector stream or not. Samples with no injection will be labeled *negative*, *i.e.* 0, and samples with an injected signal *positive*, *i.e.* 1. To avoid learned biases, the dataset will be constructed evenly, that is, half of the samples will be positive and the other half negative.

The second step deals with multiple such long-duration waveforms in a 64 s sample. This case is treated similar to an *object detection* task, where the network should predict multiple

Table 6.1: Simulation waveform parameter distributions.

Waveform parameter		Distribution
Masses	m_1, m_2	Uniform$(10\,\mathrm{M}_\odot, 80\,\mathrm{M}_\odot)$
Non-aligned spins	χ_x, χ_y	0
Aligned spins	χ_z	Uniform$(0, 0.998)$
Sky position	θ, ϕ	Sine$(0, \pi)$, Uniform$(0, 2\pi)$
Inclination and polarization	ι, ψ	Sine$(0, \pi)$, Uniform$(0, 2\pi)$
Coalescence phase	φ_{coa}	Uniform$(0, 2\pi)$
Optimal SNR	ρ	Uniform$(5, 15)$

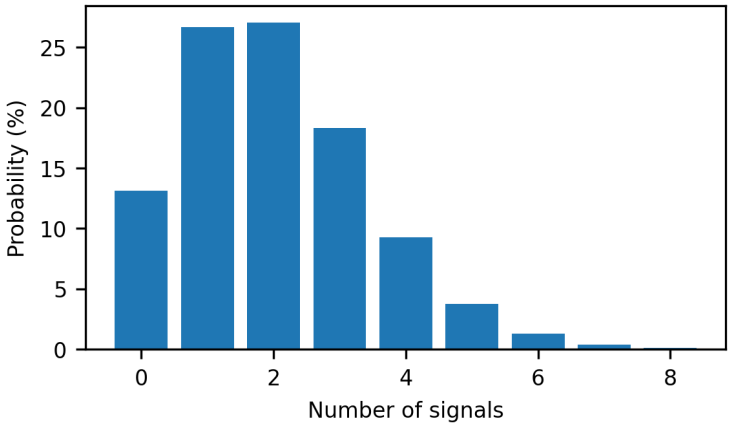


Figure 6.7: Probability of n number of signals in a 64 s sample of ET detector stream, assuming the higher event rate estimate of 10^6 yr^{-1}.

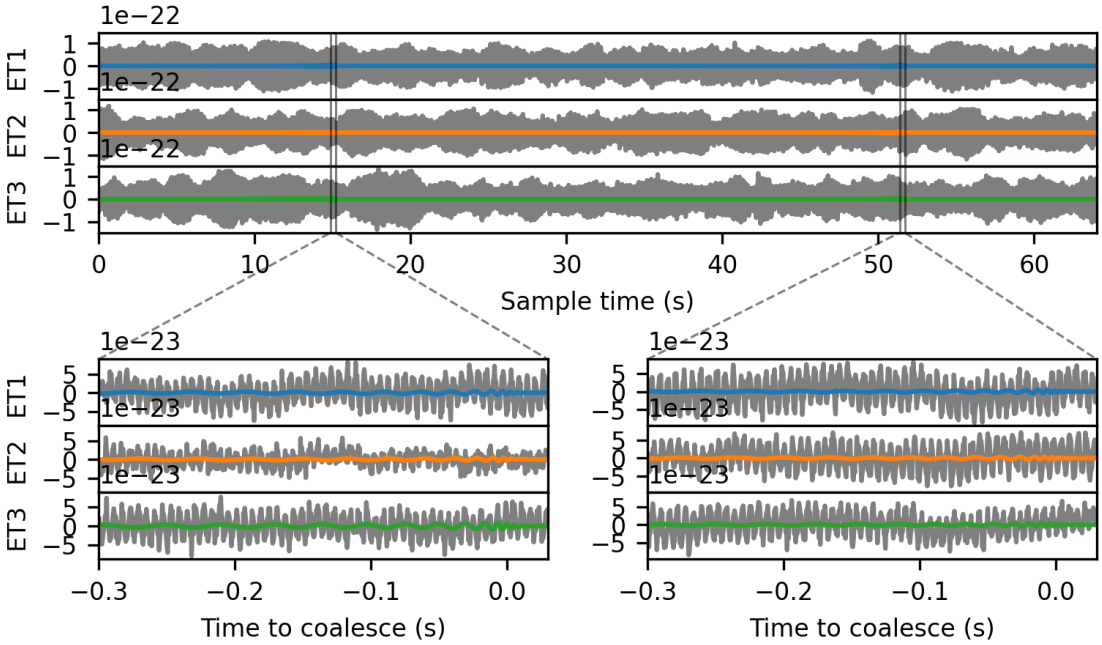


Figure 6.8: Sample with two injected signals. The total strain is plotted in gray, and the signal contribution in color. The first signal has an SNR of ≈ 6.4, and the second one has ≈ 9.1.

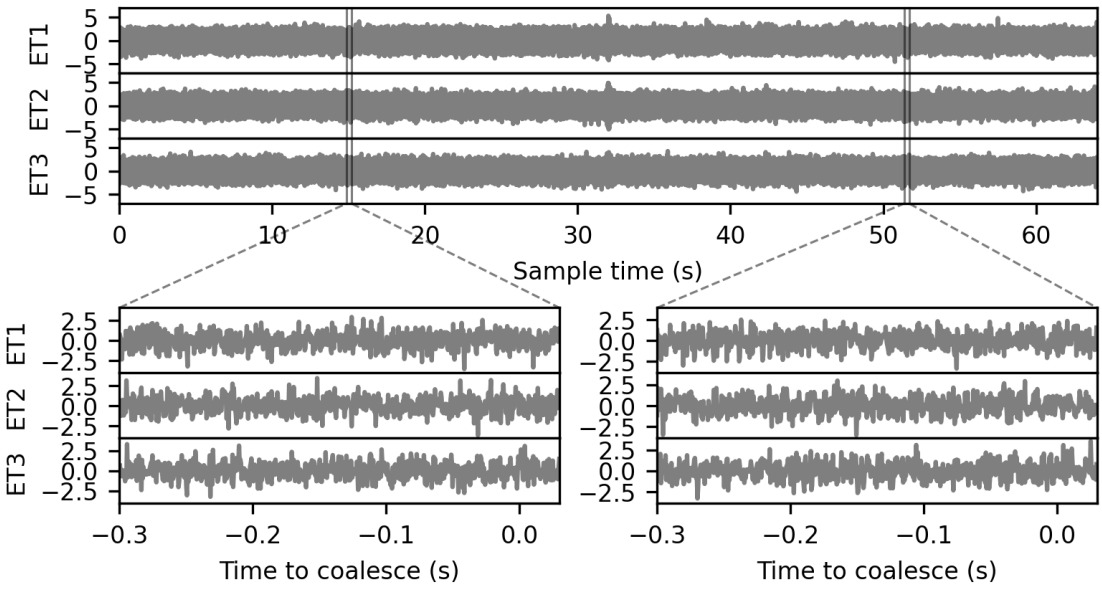


Figure 6.9: The same sample as in Fig. 6.8 after whitening, band-passing and standardizing. This resembles the data that neural networks are trained on in this work.

signals along with their coalescence time. To cover all cases evenly and ensure no biases are learned, the number signals per sample is sampled *uniformly* in $[0, 6]$. The maximum of six signals is justified by the expected event rate in ET, which is up to $\lambda = 10^6\,\mathrm{yr}^{-1}$ signals. This is a Poissonian process, thus the probability of n signals in one sample of length T is,

$$P(n) = \frac{(\lambda T)^n}{n!}\,\mathrm{e}^{-\lambda T}. \tag{6.13}$$

The distribution is visualized in Fig. 6.7. By choosing a maximum of six signals, most relevant cases are covered. See an example sample with two injected signals in Fig. 6.8. Note that this neglects the simulation of signals with a merger time outside the sample window.

6.4 Post-Processing

Three post-processing steps are performed on the samples. First, the data is *whitened*. This step aims to turn the colored noise power spectrum into a white one, which is constant over the whole frequency regime. In theory, this can be achieved by down-weighting the strain's Fourier coefficients $\tilde{s}(f)$ by the noise amplitude spectral density, *i.e.* the square root power spectral density S_n:

$$\tilde{s}'(f) = \frac{\tilde{s}(f)}{\sqrt{S_n(f)}}, \tag{6.14}$$

such that $S_n'(f) = 2\Delta f|\tilde{s}'(f)|^2 = 1$ in the absence of a signal.

In GW searches, the exact noise PSD is not known *a priori* of course, and has to be estimated from the detector stream instead. To this end, an adaption from Welch's method is used, as implemented in PyCBC [30]. This method splits the given detector strain into N overlapping segments of some duration and with a stride that is half that duration. For each segment i, the PSD is computed as,

$$S_n^i(f) = 2\Delta f|\tilde{s}^i(f)|^2. \tag{6.15}$$

Then, an estimate of the noise PSD is the *median* over all segments,

$$S_n(f) = \frac{1}{\alpha} \text{median}(S_n^1(f), S_n^2(f), \dots), \tag{6.16}$$

with a correction factor $\alpha = \sum_{i=1}^{N} (-1)^{i+1}/i$. The median instead of the mean is used to ensure that only the *noise* PSD is estimated. The latter might be biased towards larger values in the presence of gravitational wave signals.

In this thesis, the PSD is estimated for each detector and each sample separately. This roughly mimics the procedure in real searches, where the PSD estimate is adjusted constantly. Since the samples of some of the datasets are only a few seconds short, they are always simulated with some minimum length, over which the PSD is estimated, and later cut to the desired sample length. The minimum length is at least 16 s or twice the desired sample length. The extra is padded to both ends of the sample symmetrically, since the whitening procedure introduces corruption at both edges.

As another post-processing step the data is band-passed. The waveform fade-on can introduce frequency contributions below the frequency cut-off of 4 Hz, which might be a giveaway for the presence of a GW signal in the sample. Thus, the data is high-passed above 4 Hz. As this step also causes corruption at both ends of the sample, similar to whitening, the band-passing will be applied before cutting to the desired sample length.

The final processing step is standardizing the data. This is a common and often necessary data preparation step in a machine learning context, which ensures inputs with zero mean and unit variance. This is done for multiple samples at once, and for each detector i separately,

$$s_i'(t) = \frac{s_i(t) - \mu_i}{\sigma_i}, \tag{6.17}$$

where μ_i and σ_i denote the mean and standard deviation. These are computed over 4096 samples each. See a post-processed sample in Fig. 6.9.

7 Binary Classification

As previously mentioned, the task of detecting long-duration overlapping signals is split in two, focusing on the long-duration part first. In previous research, ML algorithms usually only process a short duration of data of a few seconds, which is sufficient for second-generation detectors, where signals are short [8, 9]. In contrast, the increased signal duration in ET requires processing larger amounts of data at once. This study is concerned with increasing the processed duration up to 1 min, comparable to the expected BBH signal length in ET.

To this end, multiple datasets with increasing sample lengths are simulated. For the purposes of this study, every sample contains only one signal at most. The problem of detecting multiple signals per sample is dealt with in the next chapter. This study compares a CNN, the *de facto* standard approach, and a transformer model when handling large inputs. Although [10] motivates the use of transformers for overlapping signals primarily, the transformer's ability to capture long-range dependencies could outperform CNNs in this context. The code for all machine-learning models used in this thesis is available at [51].

7.1 Problem Setup

A final sample length of 64 s is aimed for. With a sampling rate of 2048 Hz, and for three ET detectors, that means an input size of $3 \times 2048\,\mathrm{Hz} \times 64\,\mathrm{s} = 393\,\mathrm{k}$. Compare this to standard image-classification models such as ResNets [52], which handle input sizes of $3 \times 224 \times 224 = 150\,\mathrm{k}$.

This study considers datasets with sample lengths of 2 s, 4 s, 8 s, 16 s, 32 s, and 64 s. Each sample contains one signal at most, and these signals are injected uniformly over the whole sample length. The task is treated as a binary classification problem: Noise samples are labeled 0, and injection samples 1. The datasets are even, *i.e.*, half of the samples are positive and the other half negative. Training datasets have 1 M samples each, and the test datasets 131 k. Both datasets are simulated in the same manner.

It should be emphasized that this does not amount to a proper GW signal search, since binary classification becomes insufficient for long data stretches, if the signal candidate is not accompanied by its time location in that data. Still, this simplification is sufficient for isolating the problem of processing large inputs.

Binary classification models usually return a continuous output $\hat{y}$ between 0 and 1, interpreted as a signal score. To obtain a binary prediction, this score is compared to a threshold. Model output values are labeled negative if below the threshold and positive else. Depending on the target or ground truth label of a sample, four cases can be identified:

		Prediction $\hat{y}$	
		1	0
Target y	1	True positive (TP)	False negative (FN)
	0	False positive (FP)	True negative (TN)

TPs and TNs correspond to a correct classification, while FPs and FNs are errors. Evaluated over all test samples, the table above forms a *confusion matrix*.

Instead of consulting the confusion matrix for one particular threshold, it is often more insightful to plot the evolution of its entries for different thresholds. To this end, the true positive rate (TPR) and the false positive rate (FPR) are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{7.1}$$

That is, the fraction of target positive samples that are correctly identified as such, and the fraction of target negatives that are incorrectly labeled positive. Plotting the set of $(\mathrm{FPR}(z), \mathrm{TPR}(z))$ for all thresholds $z \in [0, 1]$ yields the *receiver operating characteristic* (ROC). This curve shows a model's performance independent of the chosen threshold.

There are multiple ways to compare the performance of different models. For instance, the threshold could be chosen to ensure the same FPR for all models, and the corresponding TPR could be compared. (This is the way models will be compared in the next chapter, when implementing something closer to an actual search, in which case the algorithm's false alarm rate should be low to ensure confident detections.) In this study, models are simply compared by their AUROC, the area under the ROC,

$$\mathrm{AUROC} = \int_0^1 \mathrm{TPR}\, \mathrm{dFPR}. \tag{7.2}$$

This metric returns 100 % if the classifier is perfect, while a random classifier yields 50 % AUROC.

7.2 CNN and Transformer Models

This work employs a CNN that is modeled after the *ResNet* architecture [52]. ResNets improve the traditional CNN recipe by the first introduction of residual connections. These have later also been adapted in transformers (see Ch. 5.3). Such connections ease the training of deep CNNs by allowing the redundant layers to act as identity maps.

Each convolutional block employs batch normalization [53] and uses the standard ReLU activation,

$$\mathrm{ConvBlock}(x) = \mathrm{ReLU}(\mathrm{BatchNorm}(\mathrm{ConvLayer}(x))). \tag{7.3}$$

There are residual connections every two blocks,

$$\mathrm{ResBlock}(x) = \mathrm{ReLU}(\mathrm{skip}(x) + \mathrm{ConvBlock}(\mathrm{ConvBlock}(x))). \tag{7.4}$$

If the dimensions match, *i.e.*, the convolutional blocks have stride one and the number of channels does not increase, $\mathrm{skip}(x) = x$, else, it is obtained through a convolutional operation with kernel size one.

The employed CNN is modeled after ResNet18, the smallest ResNet model with 18 layers, which is adapted to fit this task's needs (see Fig. 7.1).

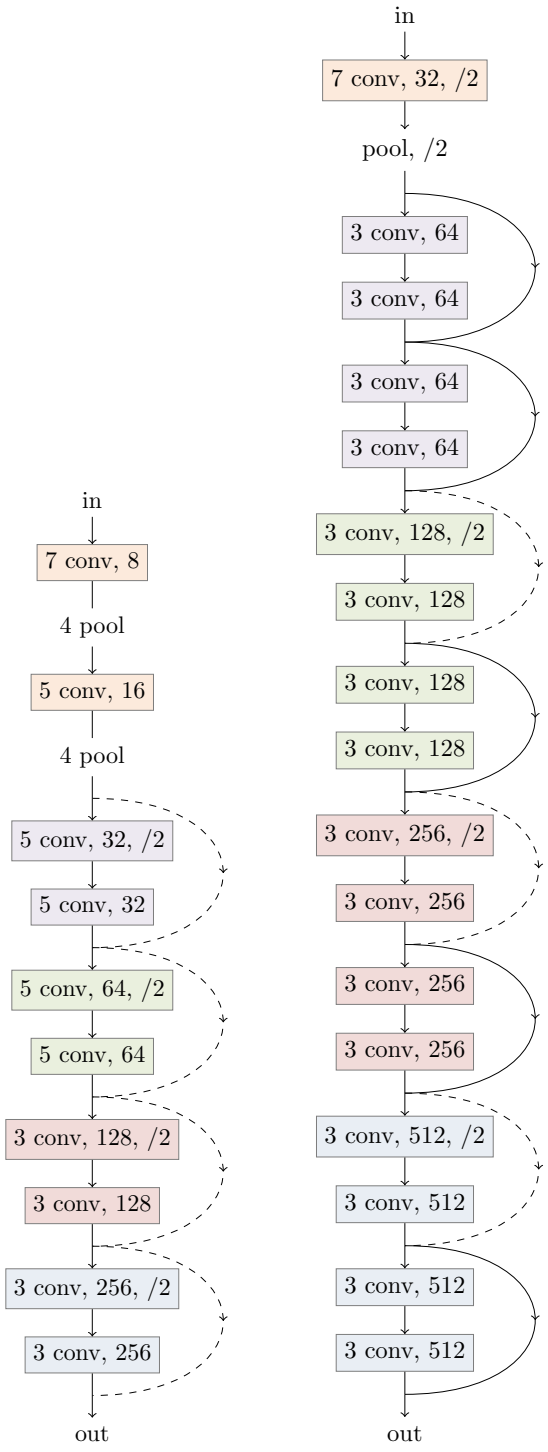


Figure 7.1: The custom ResNet architecture that is employed in this work (*left*) and ResNet18 [52] (*right*). One convolutional block is denoted "kernel size" conv, "out channels", /"stride". Skip connections are drawn as arrows. These are dashed if the dimensions do not match.

- The number of layers is reduced from 18 to 12, decreasing the total model size. The final network has 700 k trainable parameters.
- More aggressive downsampling is implemented to handle the desired long inputs. The model downsamples its input by a factor 256. For an input of size $3 \times 131\,\mathrm{k}$ (the three detectors are treated as input channels), the model produces an output of shape 256×512.
- Kernel sizes are increased to enlarge the model's receptive field. This gives a total receptive field of 1766 or roughly 1 s, which is more than enough to resolve the data's lowest frequencies of 4 Hz.

Following the ResNet architecture, *global average pool* is applied over the time dimension of the 256 feature maps. The output is fed to the classification head, a dense network,

$$h = \mathrm{Dropout}(\mathrm{ReLU}(\mathrm{Linear}_{256\to1024}(x))), \tag{7.5}$$

$$z = \mathrm{Dropout}(\mathrm{Linear}_{1024\to1}(h)), \tag{7.6}$$

producing single *logit* as output. The dropout probability is 10 %. A class score obtained through a sigmoid activation,

$$\hat{y} = \mathrm{Sigmoid}(z) = \frac{1}{1 + \mathrm{e}^{-z}}, \tag{7.7}$$

which is bounded between 0 and 1. Note that the model's number of parameters is independent of the input size. This is obviously the case for the kernel parameters, but also for the classification head, since the time information is lost after global average pooling.

The transformer model, on the other hand, is implemented similarly to the *Vision Transformer* (ViT) [54], which was originally developed for image classification. The model makes use of the transformer encoder only, which takes as input a number of tokens which are somehow obtained from the input data. The ViT architecture appends an extra learnable *classification token*. This token can attend to the input sequence through the transformer's attention mechanism and its output is read out via a classification head to yield a single class score. See a schematic of the architecture in Fig. 7.2.

In NLP, sentences can be straightforwardly tokenized into single words. For other sequential data, appropriate tokenization is less obvious, and there are multiple ways to achieve it. The original ViT model tokenizes 2D images, by splitting them into patches and projecting these onto the desired embedding dimension. For GW data, initial tests showed that this tokenization scheme is too primitive, and results in bad performance. Instead, this work opts for a hybrid approach, where a full CNN serves as the tokenizer. This is outlined in [54] as well, and a similar approach is implemented for transformer based parameter estimation of overlapping signals [10]. For this CNN tokenizer, the same ResNet model which is described above is employed without average pooling and the classification head. For the largest considered input size, this yields 512 tokens with 256 channels. The resulting tokens represent the input time series in a time-ordered manner, downsampled by a factor of 256. Each token carries the information of ≈ 1 s of data, corresponding to the CNN receptive field size. A convolution operation with kernel size one reduces the number of channels to the desired embedding dimension $d_{\mathrm{embed}} = 64$.

The classification token is a vector of the same dimension, consisting of learnable parameters. These are initialized randomly. Following [54], instead of fixed sinusoidal positional encodings, learned position embeddings are added to the input tokens. These are

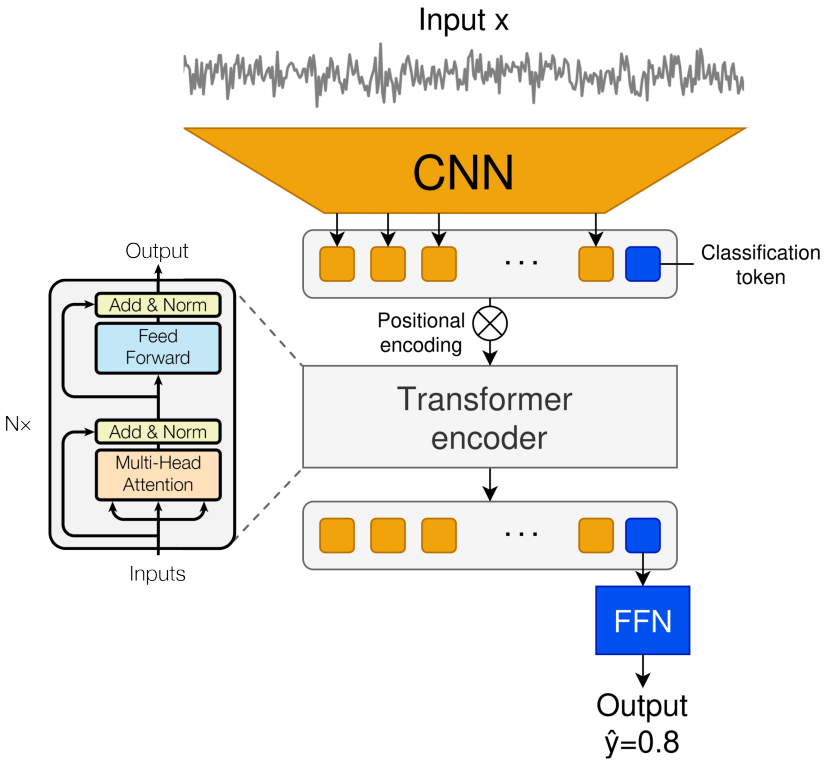


Figure 7.2: ViT architecture with an extra classification token.

initialized randomly. The transformer encoder comprises four layers, the attention blocks have eight heads each, and the dense networks' hidden dimension is of size 1024. The final classification head reads,

$$\begin{aligned} n &= \text{LayerNorm}(x), \\ h &= \text{Dropout}(\text{ReLU}(\text{Linear}_{64\to1024}(n))), \\ z &= \text{Dropout}(\text{Linear}_{1024\to1}(h)), \end{aligned} \tag{7.8}$$

which is passed through a sigmoid activation later. The model employs a dropout probability of 10 %. (It should be noted that adding a final dropout layer to the classification head is uncommon and unintentional. The results will show that this does not hinder performance, although it leads to unusual learning curves.) The total model size is 1.7 M learnable parameters. Apart from the learned embeddings, this is again independent of the input size.

7.3 Training

As is standard for binary classification tasks, the models are optimized by minimization of the *binary cross entropy* (BCE) loss. Denoting the model output $\hat{y}$ and the target label y, this reads,

$$\mathcal{L}_{\text{BCE}}(y, \hat{y}) = \begin{cases} -\log(1-\hat{y}) & \text{if } y = 0, \\ -\log\hat{y} & \text{if } y = 1. \end{cases} \tag{7.9}$$

In practice, the BCE loss is computed directly from the model *logits* z instead, *i.e.*, the unbounded pre-sigmoid outputs, with $\hat{y} = \text{Sigmoid}(z)$. This ensures numerical stability.

Training datasets are split into 90 % train and 10 % validation subsets. Models are trained for a maximum of 100 epochs with the *AdamW* optimizer [55] with a learning rate of 10^{-4}.

The optimizer employs a relatively high weight decay of 0.1 to hinder overfitting. The batch size is 64, thus each epoch comprises 15 k steps.

This work utilizes different learning-rate schedules for the two model architectures. When training the CNN, the learning rate is reduced by a factor of 10 if the validation loss has not decreased by at least 0.01 in the last 15 epochs. This is similar to [9]. The ViT is trained with a cosine-annealing learning-rate schedule [56]. This is commonly used for transformer architectures. In this case, the learning rate is adjusted by a factor α, which depends on the current training epoch and step x,

$$\alpha(x) = \begin{cases} \frac{x}{x_0} & x \in [0, x_0), \\ \frac{1}{2}(1 + \cos \pi \frac{x - x_0}{x_{\max} - x_0}) & x \in [x_0, x_{\max}), \\ 0 & \text{else.} \end{cases} \tag{7.10}$$

The schedule yields a linear increase up to $\alpha = 1$ at x_0, after which α slowly tapers off to zero at $x_{\max}$. The values x_0 and $x_{\max}$ are set to 2000 steps and 100 epochs respectively.

See the learning curves of all models and datasets plotted in Fig. 7.3. It can be seen that both the CNN and ViT models start to overfit between 20 and 40 epochs. From this, it can be concluded that both models have enough parameters for this task. Apart from the different learning schedules, differences in performance can be attributed to model architecture only. (Note that, surprisingly, for ViT, the validation loss is in fact smaller than the training loss in the early epochs. This is caused by the unintended final dropout layer, which randomly zeroes the model output with a probability of 10 %. Since dropout is only applied during training and disabled for evaluation, the model performs better on the validation dataset.)

7.4 Performance Comparison

The performance of both models with respect to dataset sample length is compared. Each model's parameters are chosen from the checkpoint with the lowest validation loss (see Fig. 7.3). The AUROC is computed over the 131 k sample test datasets, and plotted in Fig. 7.4.

Both models reach $\approx$ 88 % AUROC for the shortest input size of 2 s. Performance decreases to $\approx$ 83 % AUROC for 64 s. This decrease was expected and may be attributed to all training datasets having the same size of 1 M samples regardless of sample length. The increased size adds complexity, necessitating more training samples.

It can be seen that both the CNN and ViT perform equally well. Contrary to the hypothesis that the attention mechanism handles large inputs more efficiently, this concludes that transformers introduce no performance gain in this context. Although its memory usage is higher compared to the CNN, with at most 512 tokens for 64 s samples, it is far from the quadratic regime expected from the attention mechanism (see Fig. 7.5). This is an important intermediate result before dealing with overlapping signals.

7.5 Parameter Biases

Having established that both models can process the desired input size, the performance of the ViT model on 64 s samples is studied in more detail. The goal is to perform diagnostic checks before moving to overlapping waveforms. In particular, these are checks for any biases the model might have learned with respect to the injected waveform's parameters. Since both models perform similarly, the assumption is made that biases mostly depend on the data itself

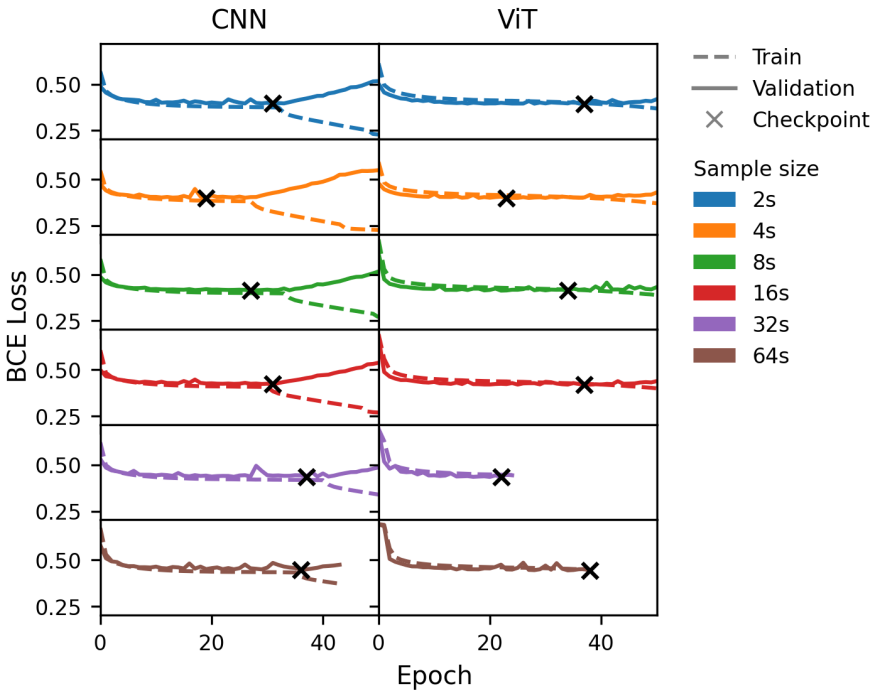


Figure 7.3: Learning curves. Evolution of the BCE train (*line*) and validation loss (*dashed*) with training epoch for the CNN (*left column*) and ViT models (*right column*). Each row corresponds to one dataset with a different sample size. A marker denotes the minimum validation loss, the point after which overfitting occurs.

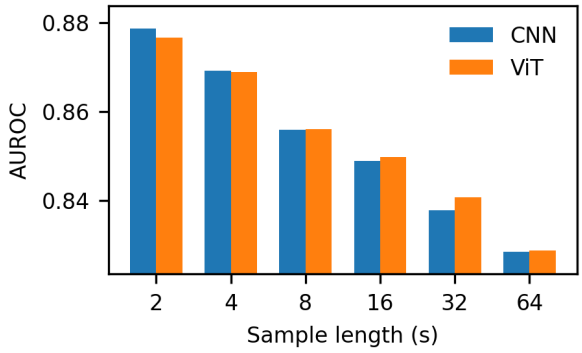


Figure 7.4: Test dataset AUROC for both models with respect to sample length. Computed over 131 k samples each.

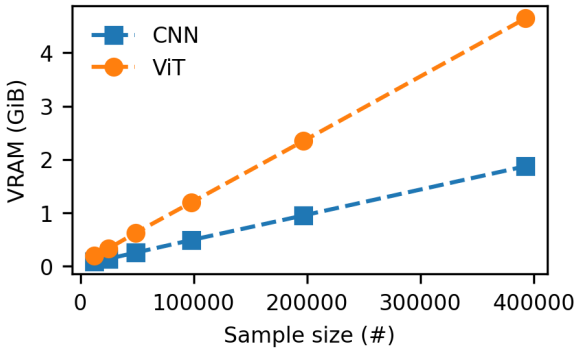


Figure 7.5: Peak training VRAM usage with input size. Measured for 64 sample batches on an RTX 4090.

and not the model type. The ViT model is chosen for this analysis because its attention weights provide an additional diagnostic of which parts of the input contribute to the classification.

Employing this model as a binary classifier requires choosing an appropriate *operating point* first. This means setting the threshold to yield some FPR and TPR. To this end, the ROC is consulted in Fig. 7.6, which shows all the possible operating points. Note that there are many points with a FPR close to zero while reaching TPR up to $\approx 50\,\%$. This reveals that a large fraction of target positive samples are easy to identify for the network. This is promising for GW searches, which focus on minimizing the false alarm rate while reporting as many signals as possible.

Again, as this study does not aim to implement a proper search, and is interested in diagnostic checks for now, the operating point can be chosen more or less arbitrarily. One such way is to find the threshold such that the resulting operating point has minimum distance to a perfect classifier with zero FPR and unit TPR. Then, the desired or optimal threshold is,

$$z_{\mathrm{opt}} = \arg\min_{z} \sqrt{\mathrm{FPR}(z)^2 + (1 - \mathrm{TPR}(z))^2}, \tag{7.11}$$

resulting in a threshold of 27.5 %. Model predictions below 27.5 % are labeled negative and positive else. This gives 17.4 % FPR and 70.2 % TPR. 17.4 % of all noise samples are incorrectly predicted to contain a signal, and 70.2 % of all injection samples are correctly identified.

The metric by which the model's performance with respect to the waveform parameters is evaluated is the *detection ratio*. That is, how many signals (which have these waveform parameters) are found? In this case, this is simply the TPR.

It is natural to first consider the model's performance with respect to SNR. Test samples are subdivided into nine SNR bins in $[5, 15]$, and the detection ratio is computed separately for each bin. The result is shown in Fig. 7.7a. In this case, half of the test dataset is considered consisting of only positive samples containing a signal. Thus, each bin consists approximately $N = 7\,\mathrm{k}$ test samples. Assuming a binomial distribution with probability p, *i.e.*, the detection ratio, then each bin has a standard deviation of $\sigma = \sqrt{p(1-p)/N}$.

It can be seen that the model achieves close to perfect detection ratios of (99.0 ± 0.1) % for the loudest signals, indicating that these signals are easily identified. For the faintest signals, the detection ratio decreases to (21.7 ± 0.5) %. This trend is expected: low-SNR signals are harder to distinguish from noise. It is also a useful sanity check, since it suggests that the network output is correlated with the signal loudness rather than with any artefacts in the simulated data.

The same analysis is applied to the signal coalescence time, i.e. the position of the merger within the sample. See the results in Fig. 7.7b. The network achieves detection ratios $\approx 70\,\%$ regardless of coalescence times. A small decrease is noticeable from (71.0 ± 0.5) % for signals whose merger lies close to the *right* sample edge down to (69.4 ± 0.5) % closer to the *left* edge. A decrease is expected, since less of the signal inspiral phase is contained in the sample in this case. Here, the effect is statistically insignificant, suggesting that predictions are made mostly based on the signals' late inspiral and merger phase, rather than by the full waveform.

Because the shape of CBC waveforms depends largely on the component masses m_1, m_2 (compare Eq. 2.20), it makes sense to investigate model performance as a function of these. To this end, both the primary and secondary masses are subdivided into seven bins in $[10\,\mathrm{M}_\odot, 80\,\mathrm{M}_\odot]$. From this, a 7×7 matrix is constructed with rows corresponding to the primary mass m_1 bins and columns to the secondary mass m_2 bins. The detection ratio is

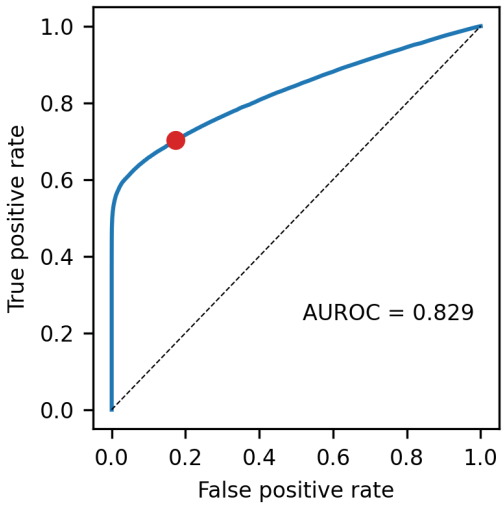


Figure 7.6: ViT ROC for 64 s samples. The dotted line shows the performance of a random classifier, which has 50 % AUROC. The red dot shows the chosen operating point.

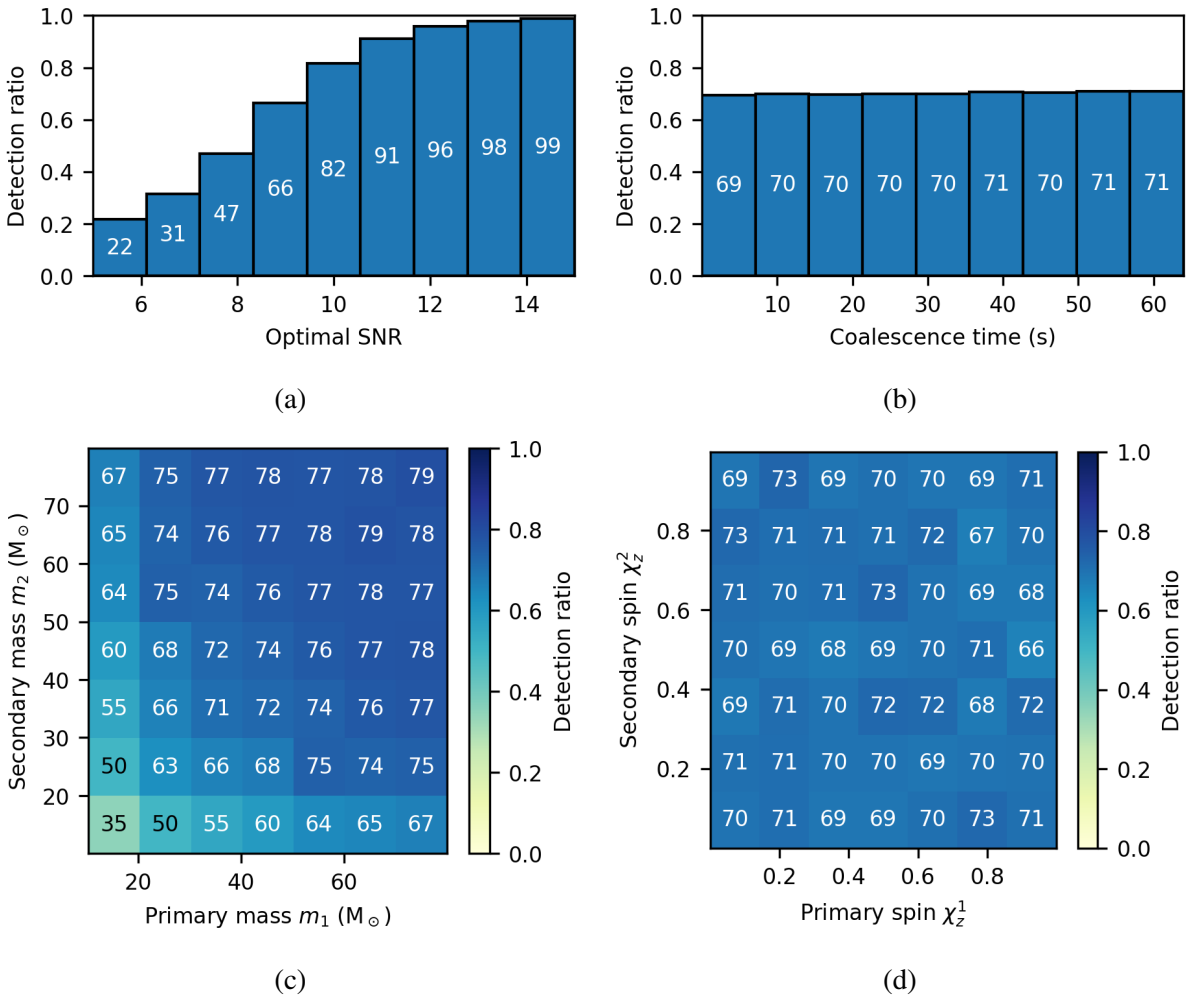


Figure 7.7: Detection ratio as a function of various signal parameters: SNR (*a*), coalescence time (*b*), the two objects masses (*c*) and spins (*d*). Computed over a total of 66 k test samples. In (*c*), the values are mirrored along the diagonal for visualization purposes. Labels are rounded to the nearest integer or the first decimal place. Uncertainties are not plotted.

computed separately for each entry. Since $m_1 > m_2$, the upper triangular half of that matrix contains no samples. To yield a full matrix, entries are mirrored along the diagonal. The result is plotted in Fig. 7.7c.

The plot exhibits two notable features, which are both statistically significant. First, the detection ratio increases with total mass, from (35 ± 1) % for the lowest total masses $M = m_1 + m_2 \approx 20\,\mathrm{M}_\odot$ up to (79 ± 1) % for the highest masses $\approx 160\,\mathrm{M}_\odot$. Second, along diagonals of approximately constant total mass, the detection ratio decreases with *mass ratio* $q = m_1/m_2$. For example, along the center diagonal with total mass $M \approx 90\,\mathrm{M}_\odot$, the detection ratio decreases from (74 ± 1) % for equal masses, *i.e.*, $q \approx 1$ to (66.5 ± 0.9) % for mass ratios close to $q \approx 8$.

Both effects can be understood from the dependence of the waveform duration on chirp mass. At leading order, the duration scales as $\tau \propto 1/\mathcal{M}^{\frac{5}{3}}$ (see Eq. 4.13). The chirp mass can be written as

$$\mathcal{M} = \frac{(m_1 m_2)^{\frac{3}{5}}}{(m_1 + m_2)^{\frac{1}{5}}} = \frac{q^{\frac{3}{5}}}{(1+q)^{\frac{6}{5}}} M. \tag{7.12}$$

Therefore, lower total masses and larger mass ratios correspond to smaller chirp masses and hence longer signals. The observed decrease in detection ratio can thus be interpreted as a degradation of model performance for longer-duration signals.

For these longer waveforms, SNR lies mostly in the long inspiral phase. If two signals have equal SNR, but one is longer, the longer waveform will have a smaller total amplitude. See Fig. 7.8 where equal-SNR waveforms with different masses are compared. Assuming sufficient template size, matched filtering detects these signals equally well, regardless of signal amplitude, since their SNR is the same. For neural networks, however, longer and lower-amplitude signals appear to be more challenging.

Finally, the detection ratio is checked against the remaining intrinsic waveform parameter, each black hole's spin z-component. Remember that only aligned-spin BBHs are considered in this work, which have vanishing spin x and y-components. Similar to the mass analysis, the primary and secondary spins are subdivided into seven bins in $[0, 0.998]$, forming a 7×7 matrix. Since the spin values are drawn independently, each entry contains 2400 samples. The result is plotted in Fig. 7.7d. A detection ratio of ≈ 70 % is achieved throughout. No clear trend is visible, and values vary between (66.1 ± 1.3) % and (72.9 ± 1.2) %, which is a difference of $3.8\,\sigma$. Thus it can be concluded that model performance is agnostic to the binary component spins. This is consistent with the earlier result that performance is mostly limited by signal length and amplitude. By simulating equal-SNR waveforms with different spins (see Fig. 7.8), it can be seen that, contrary to the effect of different component masses, the waveforms vary very little with spin.

7.6 Attention Analysis

Up to now, the results suggest that the neural network makes predictions mostly based on the waveform's merger phase where the amplitude is maximum. Because of this, performance is independent of injection time, but degrades for longer waveforms where most of the SNR lies in the inspiral part. As this analysis uses the ViT model, this hypothesis can be validated by checking the model's attention weights.

For this, a new test dataset is simulated with 66 k positive, *i.e.*, signal containing, samples with constant coalescence time. The sample size is 64 s and signals are always injected with a coalescence time of 60 s. The simulation parameters are otherwise unchanged.

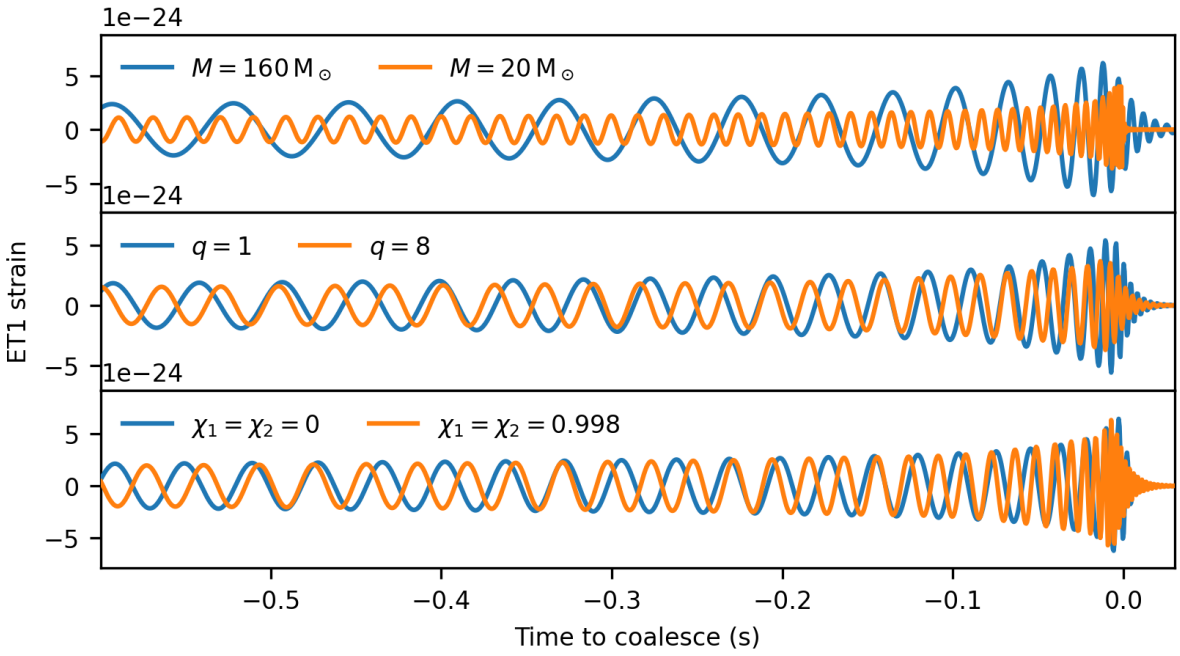


Figure 7.8: Equal-SNR ($\rho = 10$) signals with different parameters. For visualization purposes, only the ET1 channel is plotted. Each row shows two waveforms with different total mass (*top*), mass ratio (*middle*), and spins (*bottom*) and otherwise equal parameters.

To check what lead the ViT model to make its prediction, the transformer encoder weights are investigated. These correspond to a 513×513 self-attention matrix for the 512 time series tokens and the extra classification token. Each row of this matrix corresponds to one target or output token, and each entry of that row contains the weight that token assigns to the source or input tokens.

Since the final prediction only depends on the classification token, only the weights that this token assigns to the other 512 time series tokens are of interest. These weights correspond to the first 512 entries in the last row of the attention matrix. After CNN tokenization, the resulting tokens still represent the input series in a time ordered, albeit downsampled, fashion. Thus, the considered 512 classification token attention weights can be related to the input time series to show which part of the waveform is deemed important for the prediction.

In particular, the last encoder layer's weights are investigated, and these averaged over all eight heads. To gather some statistics, these weights are again averaged over multiple samples. To this end, the test dataset is divided into seven SNR bins in $[5, 15]$, each containing 9 k samples. For each subset, and for each timestep, the weights are the averaged separately. See the result in Fig. 7.9.

From this plot, it becomes clear that the attention is very *sparse*. Most attention is paid to the ≈ 1 s around the merger, leaning more towards the inspiral instead of the ringdown phase. The height of that peak grows for louder signals, from $\approx 10^{-2}$ for low SNRs up to $\approx 10^{-1}$ for high SNRs. Outside that region, the weights are smaller on average with values of order $\mathcal{O}(10^{-3})$.

This verifies that the model prediction is based mostly on the high amplitude merger phase of the signal. From this, it is clear that networks struggle with long-duration waveforms since their maximum amplitude is lower. Although the network is able to process the whole 64 s sample size, only a fraction of that information is used to make predictions.

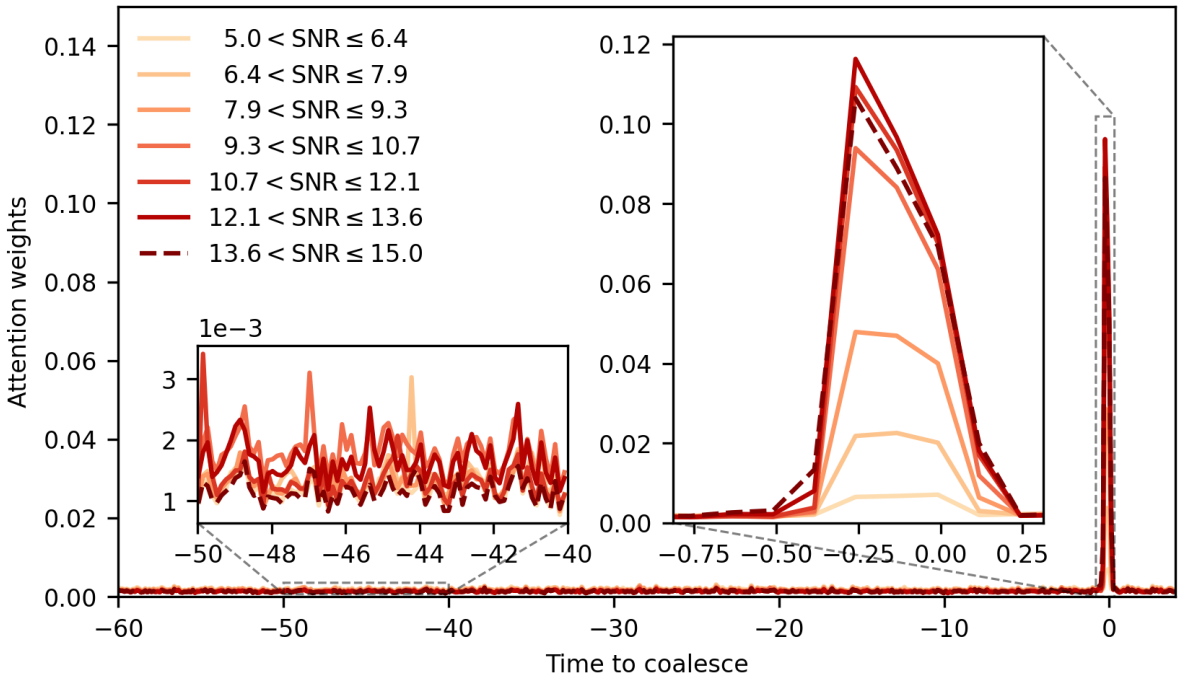


Figure 7.9: Attention weights assigned by the ViT classification token to the input time series tokens. Averaged over multiple samples with fixed merger time. Seven SNR bins are considered. The corresponding weights are plotted in different colors. Darker colors signify louder signals. The highest-SNR weights are dashed for clarity.

Again, the attention weights serve as a useful sanity check. It can be verified that the transformer is implemented correctly and that the attention mechanism works as expected. Further, the plot suggests that the network does not pick up on any possible artefacts in the data, and that predictions are based on the signal's merger phase.

7.7 Summary

This study shows that both the CNN and ViT models can process the required sample size of 64 s. A performance decrease is observed when going to larger inputs, which is attributed to training data size. No clear performance advantage of the transformer model is observed in this single-signal setting. Further analysis shows that models struggle for long-duration waveforms, such as low mass inspirals and high mass-ratio inspirals. Although the networks manage to process the full 64 s sample, they seem to focus mostly on the high amplitude merger phase, which is less pronounced in long signals.

8 Heatmap Regression

So far, this work has only dealt with the simplified problem of detecting one signal per sample, which can be treated as a binary-classification task. It remains to adapt the models to perform a proper GW search. Such a search processes a data stream of arbitrary length and returns multiple GW candidates along with some confidence measure and their time location in that data.

One simple approach, followed by the models in [57], for example, utilizes overlapping sliding windows of $\mathcal{O}(1\,\mathrm{s})$. The already trained binary classifier assigns each window a score, resulting in a "probability" time series in $[0, 1]$. From this time series, a number of post-processing steps yield multiple GW candidates.

This method is suitable for second-generation detector conditions, where signals are short. However, in ET, signal waveforms are longer and thus, larger input sizes are required of $\approx 1\,\mathrm{min}$. Further, ET's increased event rate means that these long-duration windows are almost always contaminated by some amount of signals. For example, a 64 s sample will contain the merger of at least one signal with a probability $\approx 87\,\%$ (see Fig. 6.7). Therefore, employing a classifier that was trained in Ch. 7 over sliding 64 s windows, will return mostly positive labels, making the final output useless.

An alternative approach is formulated in [9]. Here, instead of training the binary classifier separately beforehand, the network is trained to output a time series, which contains peaks corresponding to the signal coalescence times. These are then extracted through various post-processing steps to yield GW candidates. Although not stated as such in the original paper, this approach will be called *heatmap regression* in this thesis because of its similarity to an existing machine learning technique [58].

In [9], the network is trained with 8 s long samples containing one signal at most. In this work, models are trained on longer-duration samples of 64 s containing multiple signals. As transformer models have shown good performance for overlapping signals in parameter estimation [10], this chapter compares two models, one CNN based and the other transformer based. For this, minimal adjustments to the models in Ch. 7 are made, changing the task from binary classification to heatmap regression.

8.1 Problem Setup

Two datasets are simulated. The training dataset comprises 1 M samples and the test data is 131 k samples. Each sample is 64 s long and contains up to six signals with coalescence times that are sampled uniformly over the whole sample duration. The number of signals per sample is drawn from a uniform integer distribution in $[0, 6]$. The maximum is justified by the expected event rate at ET. The test dataset is simulated in the same manner as the training dataset. This includes choosing a uniform distribution for the number of signals instead of a Poissonian distribution, which would be more realistic. The dataset is not intended to be

realistic, since the SNR is also sampled uniformly instead of simulating signals to follow a distance distribution. Also, the simulation does not represent a continuous data stream. The whole evaluation should therefore be understood as a proof-of-concept study.

For each sample, the task is to make multiple GW signal predictions. These are denoted $\{(\hat{c}_j, \hat{t}_j)\}_{j=1}^{m}$, where each prediction comprises a class score $\hat{c}$ and the signal coalescence time $\hat{t}$. The ground-truth signals are characterized by their coalescence time $\{t_i\}_{i=1}^{n}$. Time values are normalized to the sample length. (Note that the number of predictions m and the number of ground-truth objects n can differ.)

When evaluating the performance of GW search algorithms, two metrics are of interest, the *false-alarm rate* (FAR) and the *detection ratio*. These are defined as,

$$\text{FAR} = \frac{\text{FP}}{\text{Length of analyzed data}}, \quad \text{Detection ratio} = \frac{\text{TP}}{\text{Number of signals in that data}}, \tag{8.1}$$

where TP and FP are true and false positives respectively. GW searches usually aim for a low FAR to make confident detections. Offline matched-filtering searches [50, 31] reach FARs $1/\text{yr}$ and online searches [29] FARs of $1/\text{month}$.

To calculate these values, it has to be considered in which case a prediction should be labeled true (TP) and in which case false (FP). It is straightforward to call a prediction TP if the class prediction exceeds a threshold, $\hat{c} > z$, and if the *corresponding* ground-truth signal is sufficiently close, $|t - \hat{t}| \leq 0.1\,\text{s}$. This is the same cutoff that is chosen in [9]. Else, the prediction is counted as a FP. Varying the threshold $z \in [0, 1]$ yields multiple operating points $(\text{FAR}(z), \text{Detection ratio}(z))$. These can be plotted as an operating characteristic, which is convenient for evaluating the performance of different models. This characteristic is analogous to a ROC curve, but uses FAR which is in units of inverse time unlike FPR which is dimensionless. Therefore, this work does not refer to these curves as ROC curves.

Now, for a given prediction, what exactly is the "corresponding" ground-truth signal? Simply taking the closest signal in time does not suffice as it can result in multiple predictions being assigned to the same signal. Instead, this task requires defining some one-to-one matching algorithm. Each signal should be assigned to one prediction, and if there are more predictions than signals, $n < m$, the left-over predictions are labeled FP.

For this, the naive approach could be adapted by iterating over all predictions and finding the closest signal under the condition that this signal has not been matched yet. Then, if prediction and signal are close enough in time, the signal is marked as matched going forward. The problem here is, that predictions and signals are assigned independent of the predicted class score. For example, a prediction with low class score can be assigned to a ground-truth signal although a prediction with high class score is also nearby. This issue can easily be resolved by iterating over predictions based on their class score, starting with the highest. The whole matching scheme will be referred to as *greedy matching*, and can be summarized as follows:

- All predictions below the threshold, $\hat{c} < z$, are noise predictions and can be disregarded.
- The remaining signal predictions are sorted by decreasing $\hat{c}$.
- For each prediction, the closest unmatched ground-truth signal is found.
- If $|t - \hat{t}| \leq 0.1\,\text{s}$, the prediction counts as a TP and the signal is marked as matched.
- Otherwise, the prediction is a FP and the signal is *not* marked as matched.
- Once there are no unmatched signals left, the remaining predictions are counted as FPs.

8.2 Heatmap Construction and Post-Processing

In heatmap regression, the model output for a sample of detector data is a time series, which is referred to as a heatmap. This heatmap consists of peaks with positions corresponding to the coalescence times of the signals in that sample. Extracting the peaks from that heatmap yields multiple GW predictions $\{(\hat{c}_i, \hat{t}_i)\}_{i=1}^{m}$. Each prediction comprises a score $\hat{c}_i$, which is taken as the peak height, and the supposed coalescence time $\hat{t}_i$, obtained from the peak location.

In order to train models to predict such a heatmap $\hat{H}(t)$, it has to be considered what the ground-truth heatmap $H(t)$ should look like. It is natural to construct the ground truth as zeros everywhere except at the exact signal coalescence times, where a label of one is chosen. From a machine-learning perspective, however, this is problematic: For one, it makes the ground truth heavily imbalanced, which complicates training. Further, the models considered here downsample by a large factor of 256, and the ground truth has to be downsampled accordingly. Sharp peaks can be lost in that process. (Alternatively, the ground truth could be downsampled with max pooling, but then some amount of time accuracy is lost.)

For these reasons, [9] employs rectangles of width 0.2 s. However, following standard heatmap regression procedure [58], this work uses Gaussian distributions instead. This choice results in better peak extraction accuracy, as the peak of a Gaussian is clearly defined, which is not the case for a rectangular distribution (see Fig. 8.1). The ground truth is constructed from Gaussians with unit amplitude and standard deviation $\sigma = 0.1$ s. The distribution is centered on the corresponding signal coalescence time t_{coa} and is truncated to a $3\,\sigma$ interval,

$$h(t) = \begin{cases} \exp\left\{\frac{-(t-t_{\mathrm{coa}})^2}{2\sigma^2}\right\} & \text{if } |t - t_{\mathrm{coa}}| \leq 3\sigma, \\ 0 & \text{else.} \end{cases} \tag{8.2}$$

From multiple such distributions, a single heatmap could be naively constructed by summing them. However, this can result in heatmap values larger than one, which is impractical for training. Instead, this work follows [60] and constructs the ground-truth heatmap using a time-wise max operation over all signal distributions,

$$H(t) = \max(h_1(t), h_2(t), \dots). \tag{8.3}$$

If a sample contains no signals, $H(t) = 0$. See an example heatmap in Fig. 8.2.

GW candidates are extracted from the predicted heatmap $\hat{H}(t)$ via a peak finding algorithm. All peaks, regardless of height, are extracted. Thresholds are applied later when computing operating points. A minimum distance of 0.1 s is enforced between peaks. For this, methods implemented by SciPy [61] are relied upon.

The extracted peak positions are denoted x_i. The GW candidate scores are the peak heatmap values, $\hat{c}_i = \hat{H}(x_i)$. For the corresponding time prediction $\hat{t}_i$, sub-pixel accuracy is achieved with a *soft argmax* operation [62], which is applied locally around peaks.

To this end, for each peak x_i, the predicted heatmap values outside a $1\,\sigma$ radius are masked. This can be written as,

$$w_i(t) = \begin{cases} \hat{H}(t) & |t - x_i| < \sigma \\ -\infty & \text{else.} \end{cases} \tag{8.4}$$

Setting masked values to $-\infty$ will be justified later on. Then, the time-prediction $\hat{t}_i$ is estimated as the soft argmax of w_i,

$$\hat{t}_i = \sum_t \mathrm{softmax}\,(w_i)(t)\, t. \tag{8.5}$$

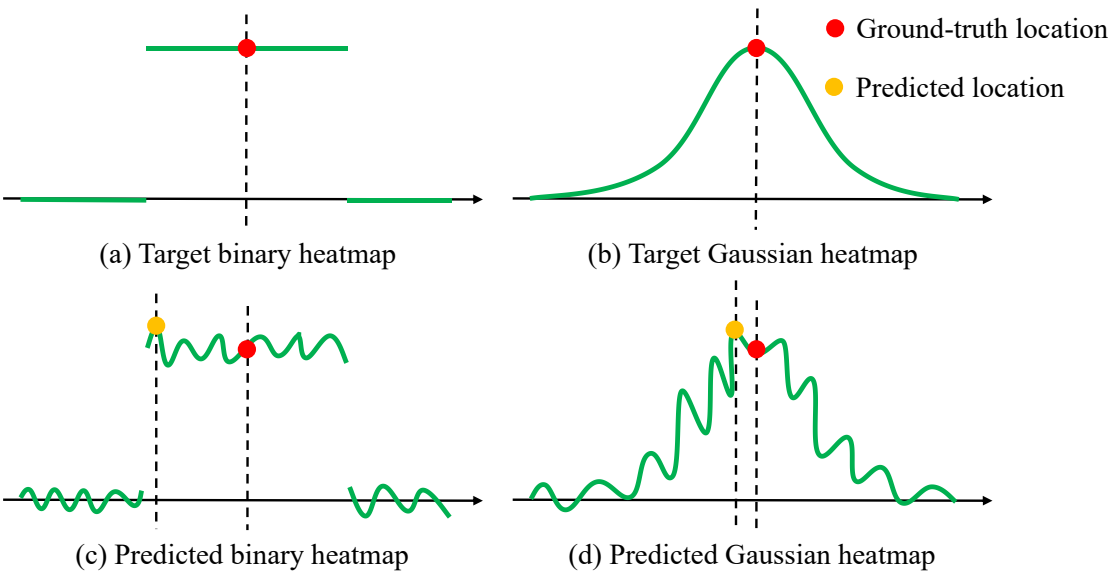


Figure 8.1: Schematics of target and predicted heatmaps. Drawn for a target rectangular distribution (*a*, *c*) and for a Gaussian distribution (*b*, *d*). Taken from [59].

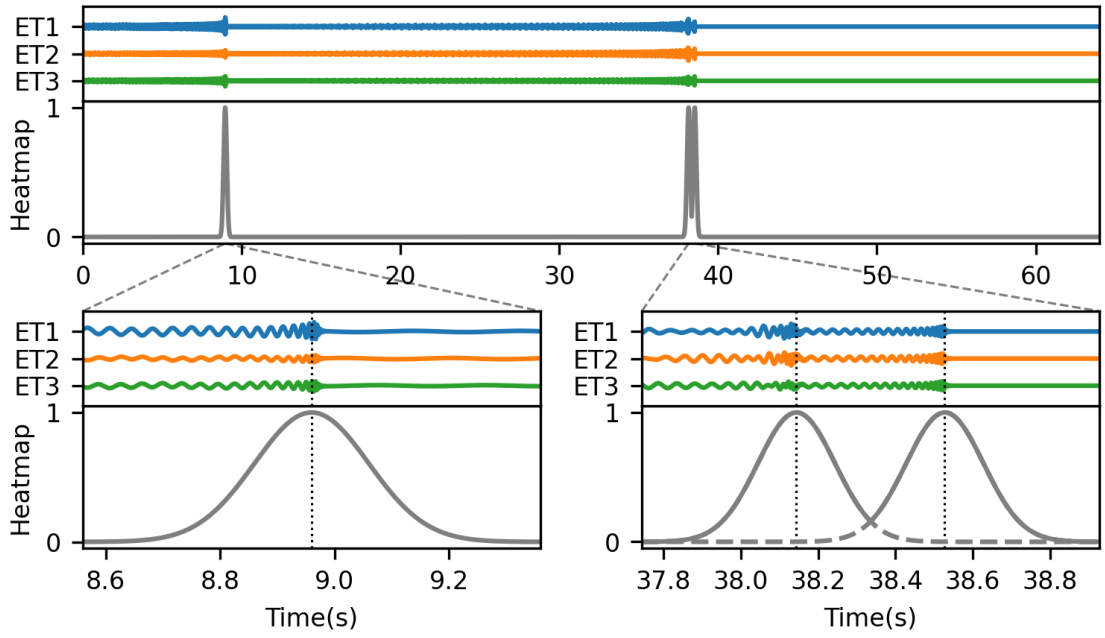


Figure 8.2: Heatmap of a 64 s sample, constructed as the time-wise maximum of multiple Gaussians. Drawn on top of each plot are the individual detector *signal* contributions. For better readability, only the signal and not the full signal + noise strain is shown.

The softmax operation,

$$\operatorname{softmax} \vec{x} = \frac{1}{\sum_i \mathrm{e}^{x_i}} \begin{pmatrix} \mathrm{e}^{x_0} & \mathrm{e}^{x_1} & \dots & \mathrm{e}^{x_{N-1}} \end{pmatrix}^\mathsf{T}, \tag{8.6}$$

ensures that the values of softmax $(w_i)(t)$ sum to one and returns zeros outside the considered $1\,\sigma$-radius, where $w_i(t) = -\infty$.

8.3 CNN and Transformer Models

Motivated by the results in [10], this chapter compares one CNN-based and one transformer-based heatmap regressor. For this, the models implemented in Ch. 7 are repurposed. The CNN model is the same ResNet model that was presented in the previous chapter. Only the average pooling over the time dimension is discarded. The classification head is then applied over all downsampled time steps to produce a heatmap of length 512. The classification head is a FFN that takes a vector with 256 features and returns one number,

$$\begin{aligned} n &= \mathrm{BatchNorm}(x), \\ h &= \mathrm{Dropout}(\mathrm{ReLU}(\mathrm{Linear}_{256\to1024}(n))), \\ z &= \mathrm{Dropout}(\mathrm{Linear}_{1024\to1}(h)), \end{aligned} \tag{8.7}$$

with a dropout probability of 10 %. The output is later passed through a sigmoid activation to yield heatmap values in $[0, 1]$. The model has a total of 700 k parameters. Apart from the exact implementation of the CNN and classification head, this architecture is the same as [9].

Similar to the ViT, the transformer model considered here consists of a transformer encoder only. The model uses the same ResNet for tokenization, again, resulting in 512 tokens with embedding dimension $d_\mathrm{embed} = 64$. Instead of appending a single classification token, *all* encoder output tokens are fed through a classification head. This is an FFN similar to Eq. 8.7 with an input size of 64 instead, and yields a heatmap of length 512. The model employs sinusoidal positional encodings, and the encoder consists of six layers and eight heads. The transformer FFNs have $d_\mathrm{ff} = 1024$. A dropout probability of 10 % is used throughout. The total number of trainable parameters is 2.2 M. See schematics of both the CNN and transformer encoder models in Fig. 8.3.

The model weights are optimized by minimization of the time-wise BCE loss of the ground truth and predicted heatmap $H(t)$ and $\hat{H}(t)$. For ground-truth labels continuous in $[0, 1]$, this is defined as,

$$\mathcal{L}_\mathrm{BCE}(H(t), \hat{H}(t)) = -(\alpha H(t) \ln \hat{H}(t) + (1-\alpha)(1-H(t)) \ln(1-\hat{H}(t))). \tag{8.8}$$

Because of label sparsity, the loss term corresponding to positive labels is up-weighted by a factor α and the other term down-weighted by $(1-\alpha)$. For this, a relatively high positive-label weight of $\alpha = 0.98$ is chosen empirically. In practice, this loss is calculated from the pre-sigmoid logits.

Models are trained for a maximum of 100 epochs using the AdamW optimizer with a learning rate of 10^{-4} and a weight decay of 10^{-4}. The batch size is 64. For the CNN model, the learning rate is reduced by a factor 0.1 if the validation loss does not decrease by at least 0.01 over 15 epochs. The encoder model is trained with a cosine-annealing schedule with a warmup of 5000 steps.

See both learning curves in Fig. 8.4. Both models start to overfit after 16 epochs. Similar to the ViT learning curves, the plot shows a lower validation loss in the early epochs because of the classification head's final dropout layer.

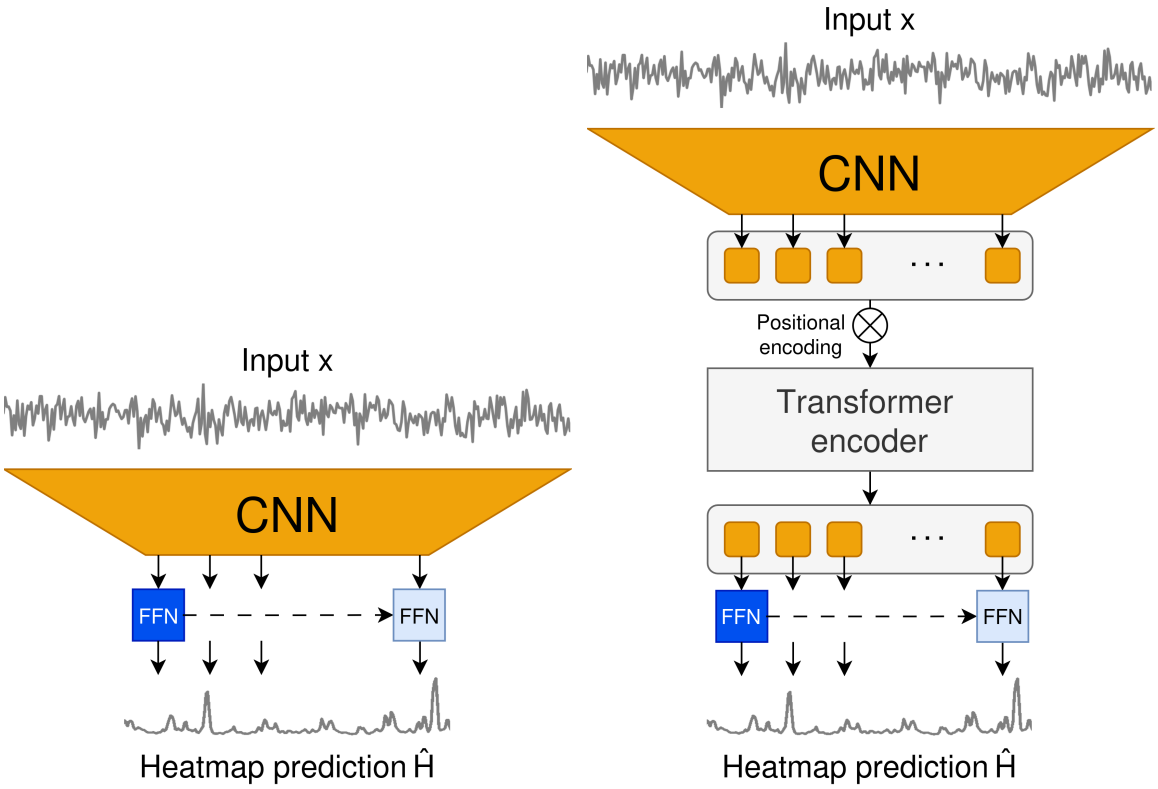


Figure 8.3: Schematics of the CNN (*left*) and the transformer encoder (*right*) heatmap regressor models.

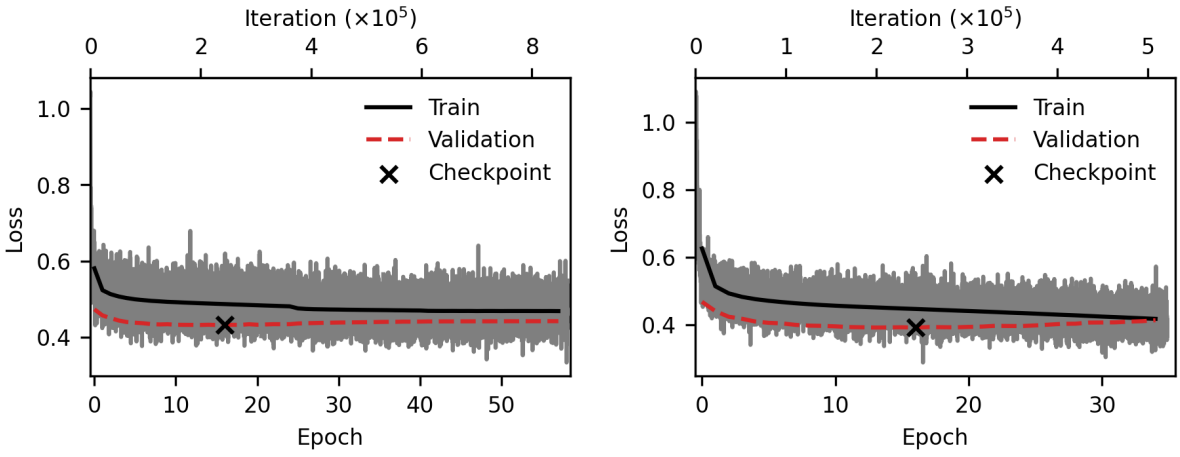


Figure 8.4: Learning curves of the CNN (*left*) and the encoder model (*right*). The training loss is plotted for each step (*grey*) and as averaged over all steps in one epoch (*black*). The validation loss is plotted in red and a marker signifies its minimum.

8.4 Performance Comparison

Before evaluating the models quantitatively, it is useful to inspect individual predicted heatmaps. Fig. 8.5 shows the transformer encoder model's output for five test samples. It can be seen that the model is quite trigger-happy, producing ≈ 100 peaks per sample. This behavior is expected to some extent, as the models were trained with a high positive weight. Still, by employing a relatively high score threshold of 96 %, most signals can be retrieved without producing too many FPs. (The exact value of that threshold depends on the desired FAR. This will be covered later.) Further, it appears that the predicted time resolution is sufficient for the chosen matching criterion: In the examples shown, no prediction is labeled FP because the time prediction differs by more than 0.1 s from the ground-truth signal coalescence time.

A robust performance test is obtained by varying the score threshold and computing the detection ratio and FAR over the whole test dataset. The detection ratio is computed over 393 k signals. This gives a maximum uncertainty of 0.08 %, assuming a binomial distribution[1]. The operating characteristics for the CNN and transformer encoder models are shown in Fig. 8.6. Both models reach low FARs while retaining a non-zero fraction of signals. At a FAR of 1/month, the detection ratio is ≈ 10 %. The encoder model performs better over the whole FAR range shown in Fig. 8.6.

To evaluate and compare the models' performance, fixed operating points have to be chosen. While low-latency searches usually aim for FARs of 1/month, the models here do not perform particularly well in this regime. For the purposes of this study, a FAR of 1/10 min is employed instead, which corresponds to approximately one FP every ten samples. At the corresponding operating points, the models achieve good performance, while not producing too many FPs. For this, the threshold is set to 97.0 % for the CNN and to 96.0 % for the encoder model. The CNN reaches a detection ratio of (61.04 ± 0.08) %, while the transformer encoder reaches (64.61 ± 0.08) %.

At this point, it is natural to ask how this performance compares to matched filtering. A direct comparison is not attempted here because implementing a full matched-filtering search goes beyond the scope of this thesis. Moreover, a quantitative comparison to existing research is complicated by the fact that the datasets used can differ significantly:

- For one, studies often employ the lower event-rates of second-generation detectors. It has been shown that for strongly time-overlapping signals, the performance of matched filtering decreases [31]. Because of ET's higher event-rate, these overlaps occur frequently in this work's simulations.
- Datasets differ in the chosen signal parameters. This matters especially for the component masses, as Ch. 7 showed that model performance depends strongly on these.
- Most importantly, datasets differ in the considered signal SNR. In this work, the SNR is sampled uniformly in $[5, 15]$. Studies often sample the distance from a second order power law instead, resulting in a non-uniform SNR distribution. For example, [31] reports SNRs up to 50, which is much louder than this work's test dataset.

For these reasons, comparisons to matched filtering will be qualitative in the following. As a rough estimate, offline searches [50, 31] reach close to perfect detection ratios for signal SNRs

[1] Strictly speaking, this assumption is wrong, as trials are not independent when dealing with overlapping signals. In particular, it will become clear that the detection ratio decreases for closely separated signals. Still, the results will show that this effect is only relevant for signals with separations ≤ 0.6 s, which does not amount to a large number of signals. Thus, the binomial uncertainty still serves as a useful estimate.

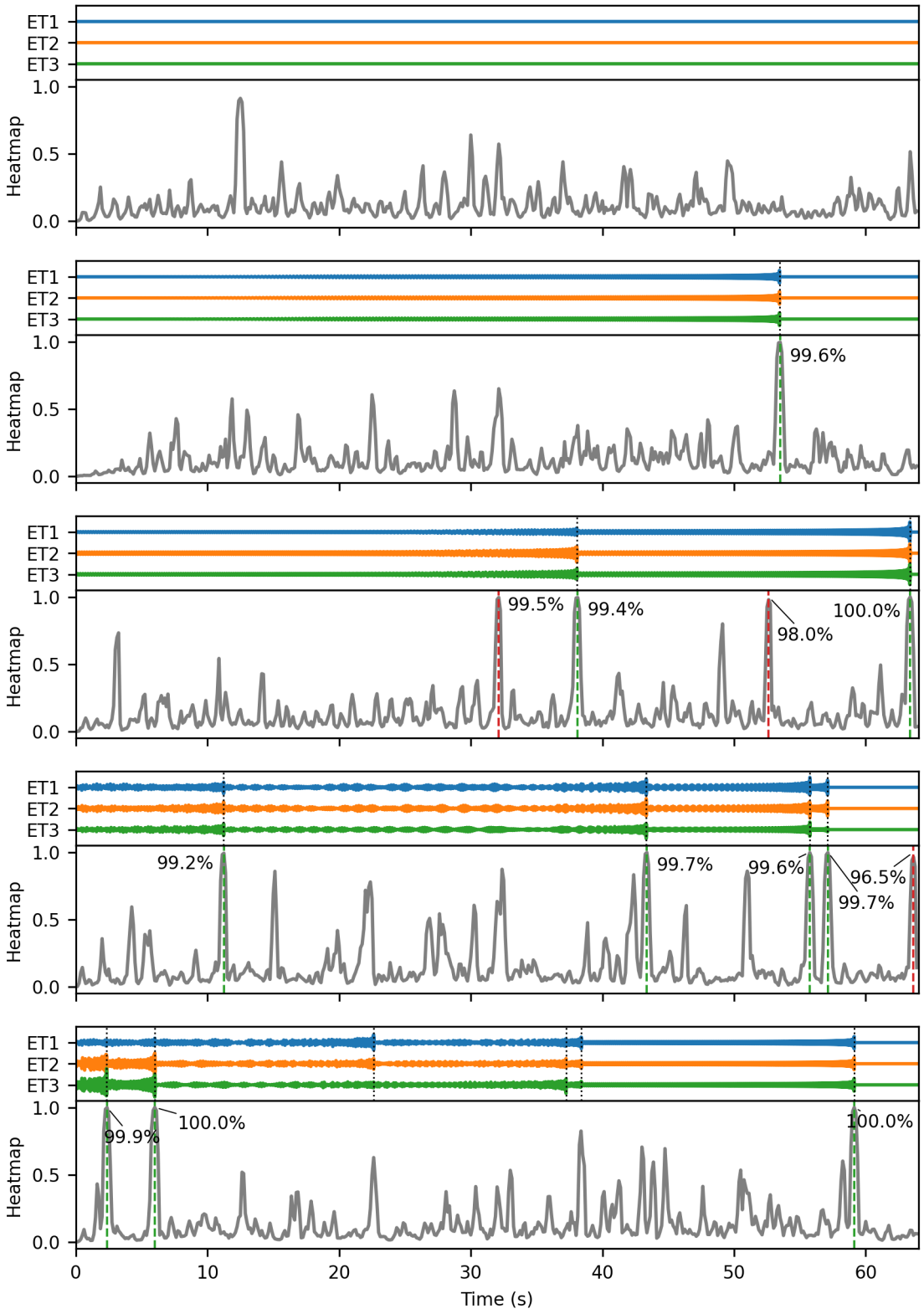


Figure 8.5: Predicted heatmaps for five samples produced by the encoder model. Peaks above a threshold of 96 % are GW candidates. For each candidate, the class score is indicated and the time prediction is shown as a vertical line. This line is colored green for TPs and red for FPs.

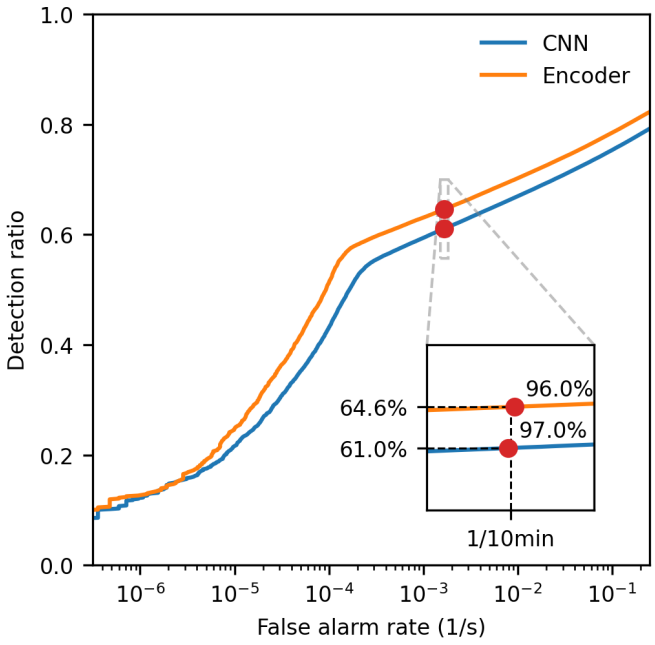


Figure 8.6: Detection ratio as a function of FAR for the CNN (*blue*) and transformer encoder (*orange*) models. Computed over 131 k test samples containing 393 k signals. The upper FAR limit corresponds to 16 FPs per sample, and the lower limit is 1/month. Markers indicate the fixed operating points, where FAR = 1/10 min. For each marker, the corresponding threshold is written out. Uncertainties are too small to be visualized.

larger than 10 to 15 and at FARs < 1/yr. For the SNR range considered here, this very loosely translates to a detection ratio of 50 %. Online searches [29] reach comparable performance at a FAR of 1/month. Compare this to the detection ratio of ≈ 10 % at a FAR of 1/month, reached by the models here.

The results so far show that the transformer model outperforms the conventional CNN approach. Motivated by [10], it is tempting to attribute this result to the increased signal overlap in the data. This can be tested by examining the detection ratio (at the fixed operating points defined earlier) as a function of the number of signals per sample n. Since the sample size is 64 s, multiple signals, $n > 1$, will almost always overlap. For this, the test dataset is binned with $n \in [1, 6]$ resulting in six subsets of 19 k samples each. These contain 19 k, 37 k, ... signals, respectively. For each subset, the detection ratio is computed separately, assuming a binomial distribution for the uncertainty. See the results in Fig. 8.7.

The plot shows that both models perform similarly if there is only one signal per sample $n = 1$. For increasing n, the CNN model stays mostly constant in performance as the detection ratio decreases by (0.3 ± 0.3) % only. Interestingly, the transformer encoder's detection ratio *increases* by (2.5 ± 0.4) % instead. Thus, the performance difference stems mainly from those samples with multiple overlapping signals: not because increased overlap degrades the CNN performance, but because the encoder improves with more signals per sample.

This difference is likely explained by the models' *receptive fields*. For the CNN, at each time step, the classification head can only attend to ≈ 1 s of data. In contrast, in the transformer encoder model, the classifier can attend to the whole input time series through the *attention mechanism*. As the CNN does not degrade in performance, it can be argued that its small local receptive field of ≈ 1 s is sufficient for capturing signal overlap. This is consistent with [31], where overlap only becomes problematic for signals with coalescence times separated by

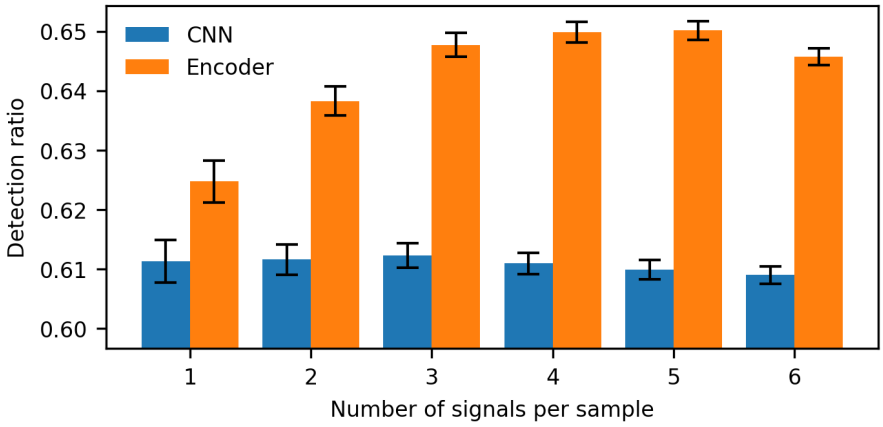


Figure 8.7: Detection ratio as a function of n, the number of signals per sample, for the CNN (*blue*) and the encoder model (*orange*). Computed over 19 k samples each. The number of signals per sample can be taken as a measure of signal overlap.

< 1 s. The effect of signal separation on model performance is studied more closely in the following sub-chapter.

How the transformer makes use of its large receptive field can be investigated by plotting the encoder attention weights. See the last layer's weights averaged over all attention heads for five samples in Fig. 8.8. These examples suggest that tokens attend to the identified signals in a sample. It can be concluded that this global information is somehow helpful for making predictions, in high-overlap conditions especially, although it is not fully clear how.

8.5 Injection Study

It has been seen that the difference in model performance between CNN and transformer encoder is associated with samples containing multiple signals. The results suggest that the CNN's ≈ 1 s receptive field is large enough to capture this overlap robustly, but the transformer's *global* receptive field appears to improve performance when dealing with multiple signals per sample $n > 1$. The effect of time overlap is isolated for the case of two signals per sample, $n = 2$. Here, according to Fig. 8.7, the transformer encoder outperforms the CNN already with a detection ratio of (63.8 ± 0.2) % compared to (61.2 ± 0.3) %. In particular, model performance is checked against the coalescence-time separation between signals Δt_{coa} in a controlled injection study. Denoting the two signals A and B, the time separation is defined as $\Delta t_{\mathrm{coa}} = |t^{\mathrm{A}}_{\mathrm{coa}} - t^{\mathrm{B}}_{\mathrm{coa}}|$.

For this, multiple 66 k sample test-datasets are simulated. What will be referred to as signal B is injected with a fixed coalescence time of 60 s, while signal A is injected at a time < 60 s. For each dataset, the signal A injection time is chosen to ensure a constant time separation. A total of eight datasets are simulated with separations of 30 s, 10 s, 3 s, 1 s, 0.6 s, 0.3 s, 0.2 s and 0.1 s. The values are chosen to be equidistant on a logarithmic scale with a higher resolution for $\Delta t_{\mathrm{coa}} < 1$ s.

For both models, the detection ratio is computed over each dataset separately. Assuming a binomial distribution, the uncertainties are < 0.14 %. See the results in Fig. 8.9. From this plot, two regimes can be identified. For $\Delta t_{\mathrm{coa}} \geq 0.6$ s, model performance is agnostic to signal separation, varying by (0.5 ± 0.2) % at most for the CNN and by (0.6 ± 0.2) % for the encoder. The transformer encoder outperforms the CNN throughout this regime by ≈ 3 %. If

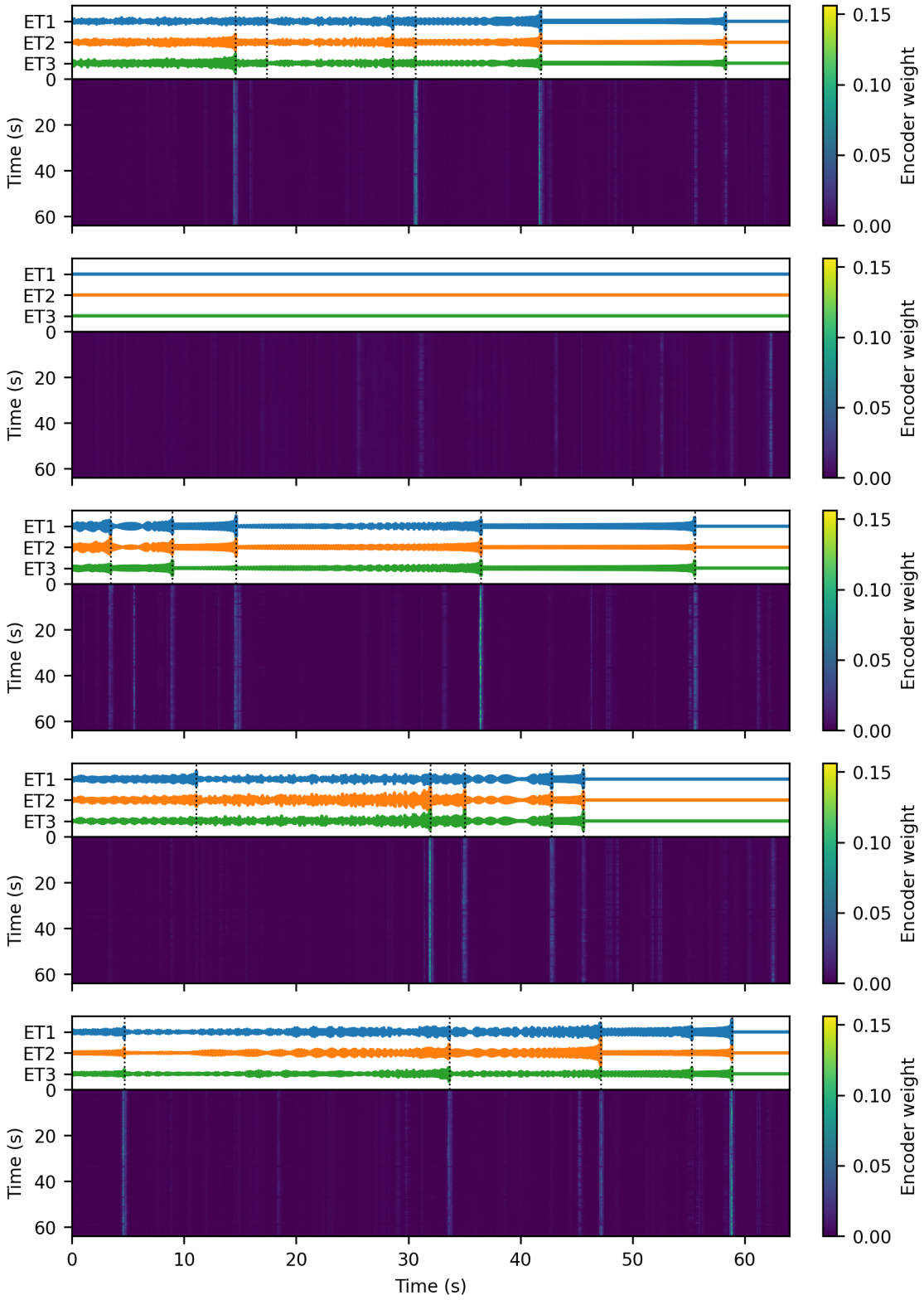


Figure 8.8: Attention matrices of the transformer encoder model for five samples. Each of the 512 tokens is assigned a time step. The x-axis shows the source sequence tokens and the y-axis the target sequence.

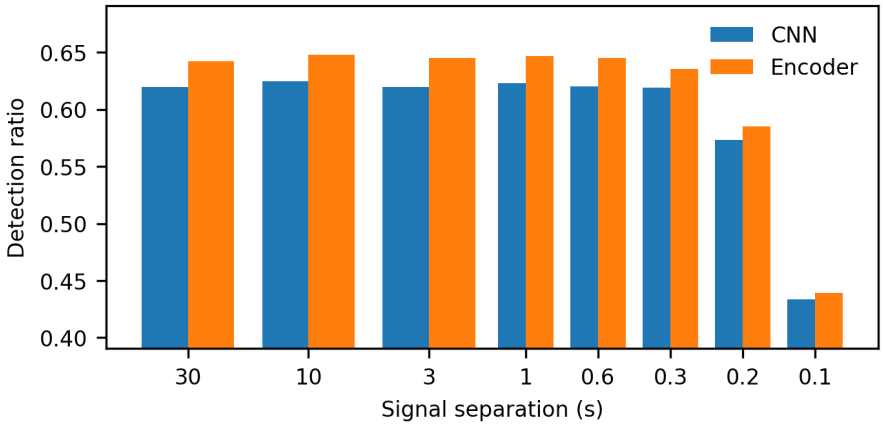


Figure 8.9: Detection ratio as a function of signal separation for the CNN (*blue*) and the transformer encoder (*orange*). Computed over 66 k samples with 131 k signals. Uncertainties are too small to be visualized.

$\Delta t_{\text{coa}} \leq 0.3\,\text{s}$, both models degrade in performance, converging to similar detection ratios of (43.3 ± 0.1) % and (43.9 ± 0.1) %.

Thus, the difference in model performance does not stem from close overlaps, but rather from long-range dependencies between signals, which are accessible to the transformer model through its global attention mechanism. For small Δt_{coa}, a similar performance decrease is noticeable in both models, suggesting a limitation shared by both architectures, which will be discussed later.

Fig. 8.9 shows that both models can detect signals with $\Delta t_{\text{coa}} \geq 0.6\,\text{s}$ with no significant performance decrease. In contrast, according to [31], standard matched-filtering searches degrade in performance for signals separated by $< 1\,\text{s}$ already. In this regime, clustering routines cause only one of the two signals to be recovered. This injection study allows for a more direct comparison with matched filtering, as the methodology is similar to [31]. See their results in Fig. 8.10. Because the exact datasets differ in signal parameters and SNRs and different FARs are employed, comparisons will be *qualitative* only.

Fig. 8.10 is reproduced for the transformer encoder by distinguishing four cases: the model finds only signal A, only signal B, both signals or neither. Because comparisons are qualitative anyway, uncertainties are simply neglected here. See the results in Fig. 8.11. The plot shows that the model can distinguish signals up to separations of 0.2 s. At $\Delta t_{\text{coa}} = 0.1\,\text{s}$, only one of two signals is recovered. Thus, at least qualitatively, the transformer encoder model is more robust than standard matched-filtering searches when handling close overlaps.

As previous results suggest a limitation in both the CNN and encoder architectures for $\Delta t_{\text{coa}} \leq 0.3\,\text{s}$, it is reasonable to attribute the performance decrease in this regime to the limited heatmap time resolution. Because of downsampling, heatmaps comprise 512 time steps only, resulting in a sample time of 0.125 s. Thus, if two signals are separated by less than *two* time steps, $\Delta t_{\text{coa}} < 0.25\,\text{s}$, the resulting heatmap prediction can only produce one peak. See example ground-truth and predicted heatmaps in Fig. 8.12.

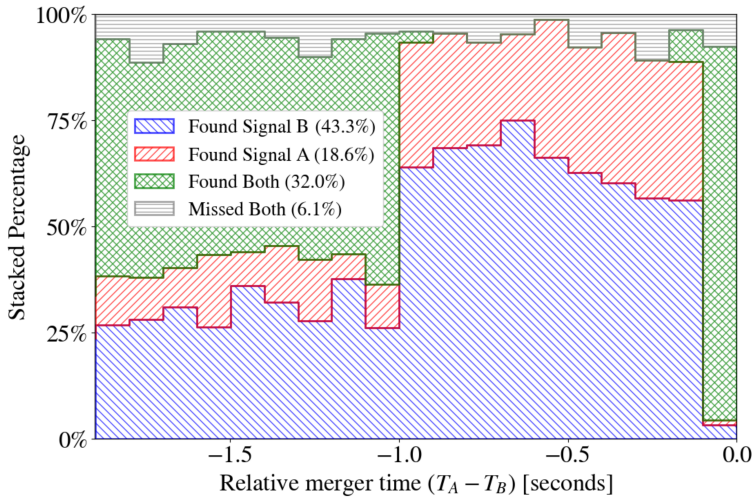


Figure 8.10: Matched-filtering performance as a function of signal separation [31]. Four cases are shown: only signal A is recovered (*blue*), only signal B (*red*), both signals (*green*) or neither (*grey*). A denotes the earlier signal.

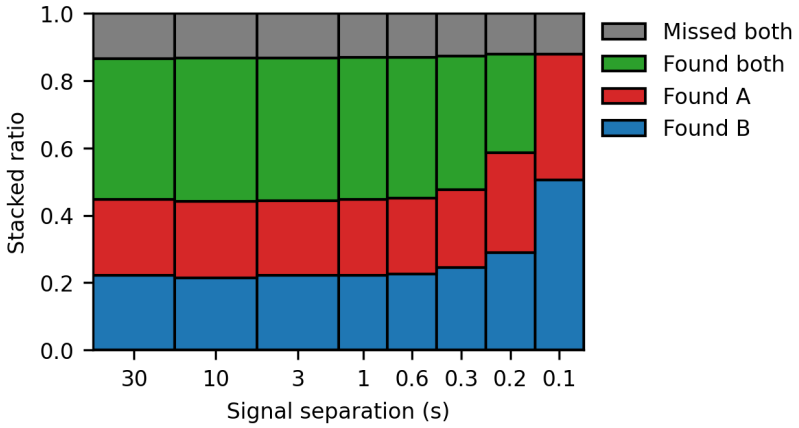


Figure 8.11: Transformer encoder performance with signal separation. For comparison with Fig. 8.10, notice the different x-axis scale.

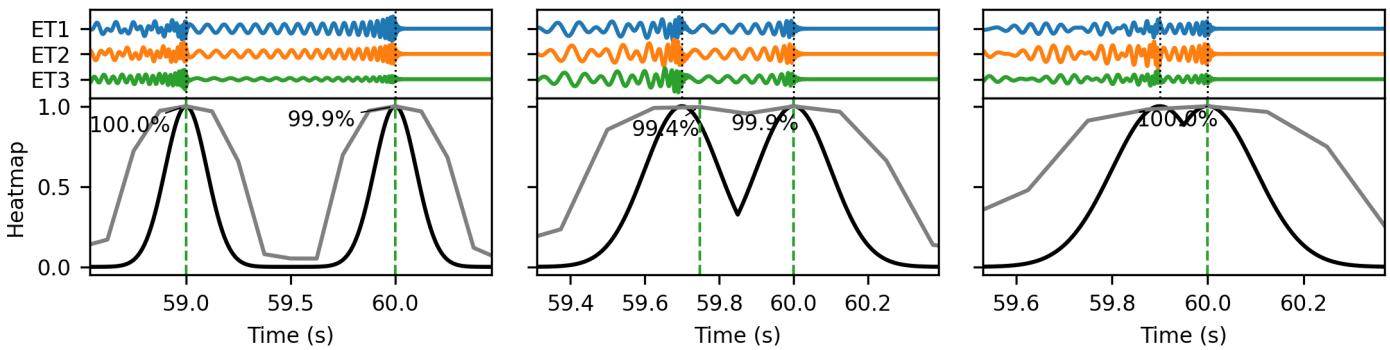


Figure 8.12: Three example ground-truth (*black*) and predicted heatmaps (*grey*) of the transformer encoder model for $\Delta t_{\text{coa}} = 1\,\text{s}$ (*left*), $\Delta t_{\text{coa}} = 0.3\,\text{s}$ (*middle*) and $\Delta t_{\text{coa}} = 0.1\,\text{s}$ (*right*).

8.6 Summary

This chapter reformulated GW detection as a heatmap regression task. Adaptations have been made to the approach in previous research [9] to reduce the number of post-processing steps and to handle multiple signals per sample.

Two architectures were investigated: a CNN and a transformer encoder, which were adapted from the binary-classification models introduced previously. Both achieved low FARs down to 1/month while retrieving a non-zero fraction of signals, with the transformer consistently outperforming the CNN.

An injection study showed that this advantage does not arise from resolving closely overlapping signals, but from exploiting long-range dependencies between multiple signals. Both architectures exhibited a common performance limit for signal separations $\leq 0.3\,\mathrm{s}$, consistent with the resolution of the predicted heatmaps.

9 Detection Transformer

The previous chapter showed that heatmap regression performs well for GW searches. However, the approach is limited by the time resolution of the output heatmap, which becomes problematic for close overlaps. One possible solution would be to increase the heatmap resolution, for example by adding *upsampling* steps after the transformer encoder, similar to U-Net architectures [63].

In this chapter, a different approach is investigated, forgoing the heatmap approach entirely. Instead, a fully end-to-end model is developed: for a given sample, the model directly outputs multiple GW candidates, each with a signal score and time location. The approach is based on the Detection Transformer (DETR) [64], which was originally developed for object detection in images and is modified here to handle time series instead. Although not implemented in this thesis, such a model can simply be adapted to return an estimate of the signal parameters as well, which is important for low-latency searches.

9.1 Architecture

The DETR architecture consists of a CNN backbone, a transformer encoder, and a transformer decoder. Up to the encoder, the architecture is similar to the transformer-based heatmap-regressor. The difference is that the encoder output tokens are not read out directly to produce a heatmap, but serve as the source sequence of a transformer decoder. The decoder takes a number of learned *queries* as its target sequence. The decoder output is read out with two prediction heads to produce a class score and the signal coalescence time. In principle, by appending more prediction heads, this architecture can be adapted to return other parameters as well. See a schematic of the original architecture in Fig. 9.1. In a way, DETR is a natural extension of the ViT [54] architecture to multiple classification tokens.

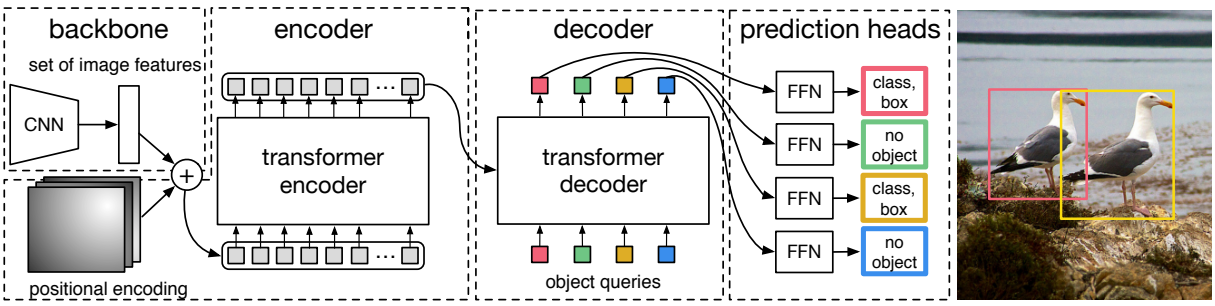


Figure 9.1: Original DETR [64] architecture for image object detection. The model outputs multiple object class scores and bounding boxes.

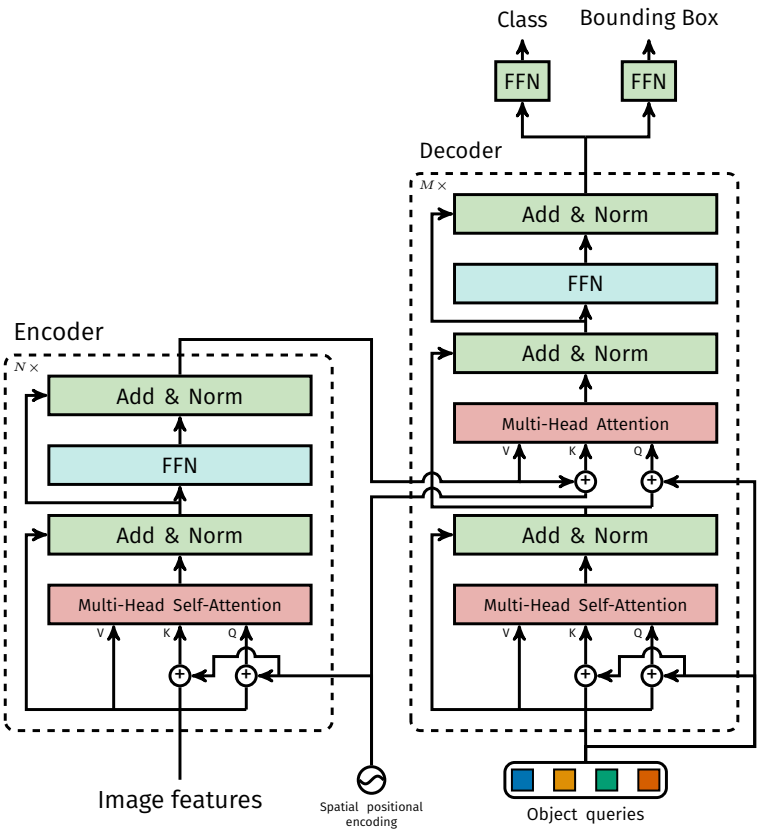


Figure 9.2: Detailed implementation of the transformer used in DETR [64]. Positional encodings are not applied before the value projections.

One limitation of the DETR architecture is that the number of possible predictions per sample is fixed by the query count. This problem can be dealt with by simply choosing the number of queries to be sufficiently high.

For the CNN backbone, the same ResNet model as previously is employed. Given samples with a length of 64 s, the backbone outputs 512 vectors of size 256. A convolution operation with kernel size one projects these onto the embedding dimension $d_{\text{embed}} = 64$. Both the transformer encoder and the decoder consist of six layers and eight heads, and employ FFNs with hidden dimension $d_{\text{ff}} = 1024$. A dropout probability of 10 % is used throughout the model. The transformer employs sinusoidal encodings for the encoder tokens and learned positional embeddings for the queries. Following the original architecture, these are applied before the query and key projections only (see Fig. 9.2).

A total of 16 queries is chosen. For the higher event-rate estimate of $\lambda = 10^6\,\text{yr}^{-1}$, the probability for more than 16 signals in a $T = 64\,\text{s}$ long sample is negligible:

$$P(n > 16) = 1 - \sum_{k=0}^{16} \frac{(\lambda T)^k}{k!}\, \mathrm{e}^{-\lambda T} = 7 \times 10^{-9}\,\%, \tag{9.1}$$

making this choice more than sufficient. These queries are learnable parameters of size d_{embed}, which are initialized as zero. After being passed through the decoder, each query is read out via two prediction heads. The class score is obtained through,

$$\begin{aligned} n &= \text{LayerNorm}(x), \\ h &= \text{ReLU}(\text{Linear}_{64\to1024}(n)), \\ z_c &= \text{Linear}_{1024\to1}(h), \end{aligned} \tag{9.2}$$

which is later passed through a sigmoid layer. For the time prediction, the prediction head reads,

$$\begin{aligned} n &= \text{LayerNorm}(x), \\ h_1 &= \text{ReLU}(\text{Linear}_{64\to1024}(n)), \\ h_2 &= \text{ReLU}(\text{Linear}_{1024\to1024}(h_1)), \\ h_3 &= \text{ReLU}(\text{Linear}_{1024\to1024}(h_2)), \\ z_t &= \text{Linear}_{1024\to1}(h_3). \end{aligned} \tag{9.3}$$

This value is also passed through a sigmoid layer to normalize the prediction to $[0, 1]$ in units of sample duration. In total, the network has 6.2 M trainable parameters.

9.2 Training

For any given sample, the DETR architecture implemented here produces $m = 16$ predictions $\hat{y} = \{(\hat{c}_j, \hat{t}_j)\}_{j=1}^{m}$, each comprising a signal score and coalescence time. The network is of course free to output predictions with $\hat{c} = 0$, *i.e.* noise predictions, which are discarded after thresholding. The class score and time prediction are trained in tandem, employing a BCE loss and an L_1 loss, summed over all predictions:

$$\mathcal{L}(y, \hat{y}) = \sum_{j=1}^{m} \mathcal{L}_{\text{BCE}}(c_j, \hat{c}_j) + c_j \, \lambda |t_j - \hat{t}_j|. \tag{9.4}$$

For predictions where the ground-truth class is noise, $c = 0$, the time prediction loss is simply zeroed out. Since for random predictions the average BCE loss is $\mathcal{L}_{\text{BCE}} = \ln 2 \approx 0.69$, while the L_1 loss to the nearest signal is only of order 10^{-2}, the L_1 term is up-weighted by a factor $\lambda = 6$. The exact value is chosen empirically. The architecture makes use of *auxiliary* losses, where the loss is computed over the output at each of the six decoder layers.

Up to this point, it has been quietly swept under the rug what the ground truth $y = \{(c_j, t'_j)\}_{j=1}^{m}$ corresponding to the m predictions actually looks like. For this purpose, all signals $\{t_i\}_{i=1}^{n}$ (where $n < m$) are assigned to predictions by a matching algorithm. Then the ground truth is constructed with

$$y_j = (c_j, t'_j) = \begin{cases} (1, t_i) & \text{if signal } i \text{ is matched to prediction } j, \\ (0, \varnothing) & \text{else.} \end{cases} \tag{9.5}$$

For unmatched predictions, the ground-truth coalescence time does not matter as the corresponding L_1 loss is zeroed out anyway.

Following [64], signals i and predictions j are matched by minimizing the cost,

$$\mathcal{L}_{\text{cost}}(t_i, \hat{y}_j) = -\hat{c}_j + \lambda |t_i - \hat{t}_j|, \tag{9.6}$$

where again $\lambda = 6$. Let $\tau : \mathcal{I} \to \mathcal{J}'$ denote an *assignment function* mapping the signal indices $\mathcal{I} = \{1, 2, \ldots, n\}$ to the subset of matched prediction indices $\mathcal{J}' \subset \mathcal{J} = \{1, 2, \ldots, m\}$. The *optimal* assignment function π will be the realization of τ that minimizes the total assignment cost over all signals i and corresponding predictions $\tau(i)$:

$$\pi = \arg\min_{\tau} \sum_{i=1}^{n} \mathcal{L}_{\text{cost}}(t_i, \hat{y}_{\tau(i)}). \tag{9.7}$$

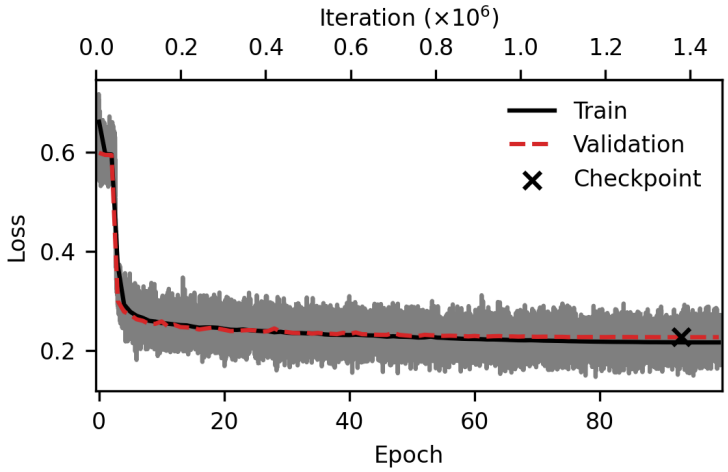


Figure 9.3: DETR learning curve. Total loss computed over the training dataset after each step (*grey*) and averaged over each epoch (*black*). Validation dataset loss (*red*). The minimum validation-loss checkpoint is indicated.

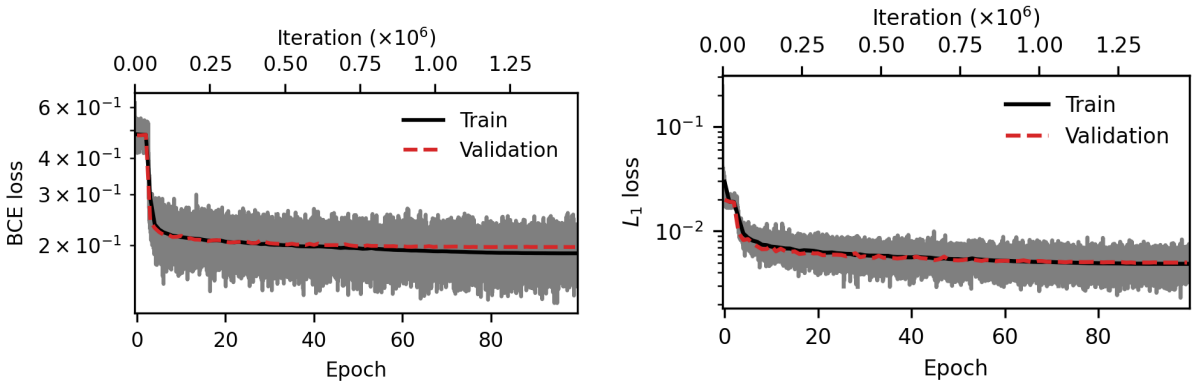


Figure 9.4: Evolution of both loss terms during training. BCE loss (*left*) and time prediction L_1 loss (*right*).

This is referred to as an *assignment problem* and can be solved with methods implemented by SciPy [61].

It can be seen that the index i of a signal that is matched to a prediction j is given by $\pi^{-1}(j)$. This function $\pi^{-1}(j)$ is only defined for those predictions j, which have been matched, $j \in \mathcal{J}'$. Thus with Eq. 9.5, the ground truth y_j that corresponds to $\hat{y}_j$ can be constructed as,

$$y_j = \begin{cases} (1, t_{\pi^{-1}(j)}) & \text{if } j \in \mathcal{J}', \\ (0, \varnothing) & \text{else.} \end{cases} \tag{9.8}$$

To differentiate this scheme from greedy matching, which is used for evaluation, the matching algorithm here will be referred to as *Hungarian matching* following the terminology in [64].

The DETR model is trained on the same 1 M sample dataset as the heatmap models. The AdamW optimizer is used with a learning rate of 10^{-4}, a weight decay of 10^{-4} and a batch size of 64. Training is performed for a maximum of 100 epochs. For this, a cosine-annealing learning-rate schedule is employed with a warmup of 5000 steps.

See the total loss learning curve in Fig. 9.3. The validation loss reaches its minimum after 93 epochs. This checkpoint is used for all following results. Fig. 9.4 shows the evolution of the individual loss terms. It can be seen that both the class score and the time prediction have been successfully optimized in tandem.

9.3 Comparison with Heatmap Regression

The following results and discussion largely mirror the structure of the previous chapter. The DETR model is evaluated on the same 131 k-sample test dataset, and GW candidates are assigned to signals with greedy matching. See the methodology in Ch. 8.1.

It is again insightful to first inspect individual model outputs. See five examples in Fig. 9.5. It can be seen that the DETR queries produce meaningful GW candidates. By applying a class-score threshold of 69 %, most signals can be retrieved while keeping the number of FPs small. The choice of this threshold is justified later on. The predicted coalescence times also appear sufficiently accurate for the matching criterion $|t - \hat{t}| \leq 0.1$ s used in this work.

The DETR prediction queries can attend to the time series through cross-attention. The plots in Fig. 9.5 indicate that each query focuses mostly on a localized region of the input, close to the corresponding signal coalescence time.

A quantitative comparison is obtained by varying the score threshold and computing the detection ratio and FAR over the full test dataset. Figure 9.6 compares the resulting operating characteristic of DETR with both heatmap regressors from the previous chapter. Over a large part of the FAR range, DETR performs comparably to the transformer heatmap model. For FARs from $\approx 10^{-4}\,\mathrm{s}^{-1}$ down to values between $10^{-5}\,\mathrm{s}^{-1}$ and $10^{-6}\,\mathrm{s}^{-1}$, DETR even reaches a higher detection ratio. At the lowest FARs, however, DETR falls below both heatmap regressors.

At the fixed FAR operating point of 1/10 min, DETR reaches a similar detection ratio of (64.90 ± 0.08) %, compared to (64.61 ± 0.08) % for the transformer-encoder heatmap regressor. The corresponding score threshold is 69.0 %. Thus, DETR achieves comparable performance to heatmap regression while directly producing candidate scores and times, without requiring any post-processing steps.

The dependence on the number of signals per sample is shown in Fig. 9.7. For $n = 1$, DETR reaches comparable performance to the other models. Similar to the transformer-encoder heatmap regressor, DETR improves for larger n as the detection ratio increases by (3.4 ± 0.4) %.

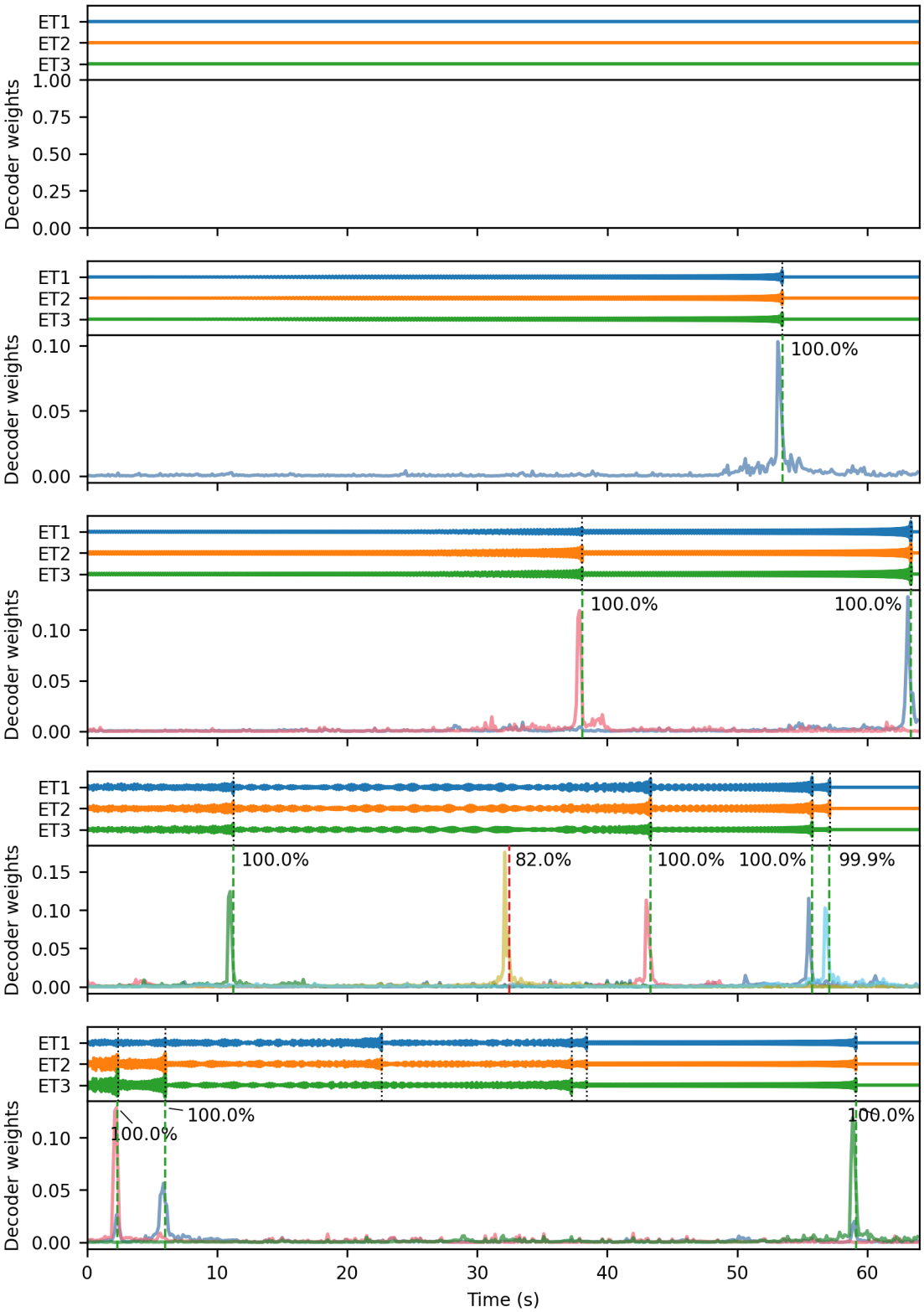


Figure 9.5: DETR output for five samples. Each query is a GW candidate. These are plotted in the same manner as in Fig. 8.5, and only candidates with scores exceeding 69 % are shown. For each prediction, query–time cross-attention is plotted in different colors.

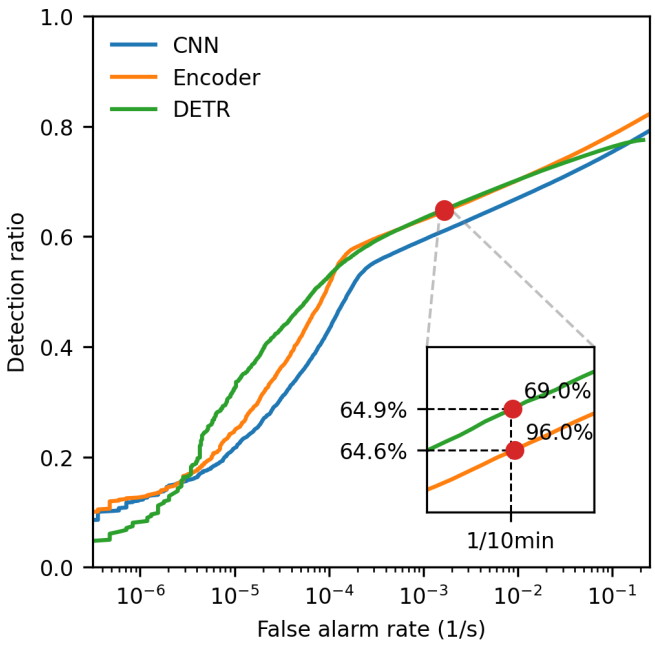


Figure 9.6: Detection ratio as a function of FAR for the CNN heatmap regressor (*blue*), transformer-encoder heatmap regressor (*orange*), and DETR model (*green*).

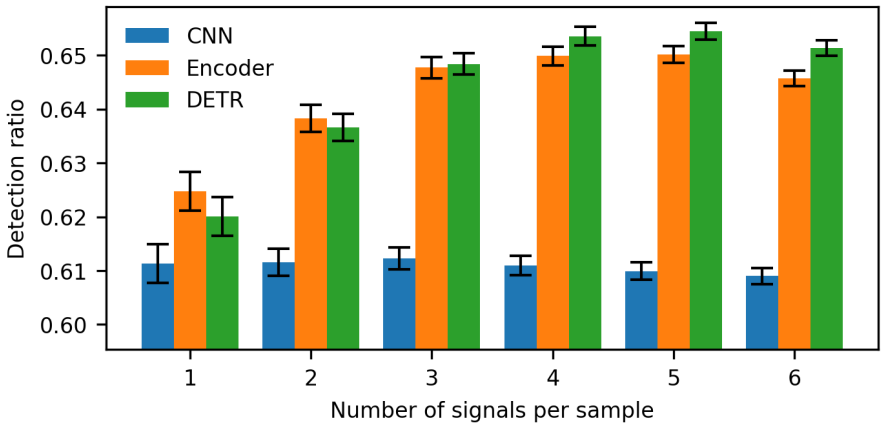


Figure 9.7: Detection ratio as a function of the number of mergers per sample n. CNN heatmap regressor (*blue*), transformer-encoder heatmap regressor (*orange*), and DETR model (*green*).

This is consistent with DETR's use of global attention, which already appeared advantageous in the transformer heatmap regressor.

An important difference between DETR and heatmap regression is the way coalescence times are predicted. In heatmap regression, predictions are obtained by peak extraction and a local soft-argmax, while DETR predicts coalescence times directly. The effect on prediction time-accuracy is investigated by computing the error $\hat{t} - t$ for all TP predictions. The errors are plotted as histograms for both models in Fig. 9.8.

Both error distributions have zero mean. For DETR, the distribution is sharper with a standard deviation of 17 ms compared to 34 ms for the heatmap regressor. Thus, DETR improves the time-prediction accuracy.

9.4 Injection Study

The previous chapter showed that heatmap regression becomes limited for nearby signals with $\Delta t_{\text{coa}} \leq 0.3$ s, as the two heatmap peaks can no longer be distinguished. Since DETR predicts coalescence times directly, one might expect improved performance for close overlaps. To test this, the same two-signal injection study as in the previous chapter is repeated for DETR.

Fig. 9.9 shows the detection ratio as a function of coalescence-time separation. The plot shows that, for separations ≥ 3 s, both models perform equally well. DETR exhibits a performance decrease for separations of ≤ 1 s compared to ≤ 0.3 s for the heatmap regressor. However, this decrease is less pronounced for close separations of 0.1 s, where DETR retains a detection ratio of (59.0 ± 0.1) % compared to (43.9 ± 0.1) %.

A more detailed view is obtained by distinguishing the four recovery cases: only the earlier signal A is recovered, only the later signal B, both signals are recovered, or both signals are missed. The result is shown in Fig. 9.10, together with the corresponding heatmap-regression result. Three example outputs are shown in Fig. 9.11 as well.

DETR can still output two GW candidates for separations of ≤ 1 s, but the fraction of samples where both signals are recovered decreases in the close-overlap regime. Instead, single-signal recoveries become more common. This suggests that, while DETR *can* make predictions regardless of signal coalescence-time separation, it is not perfectly robust when disentangling close overlaps.

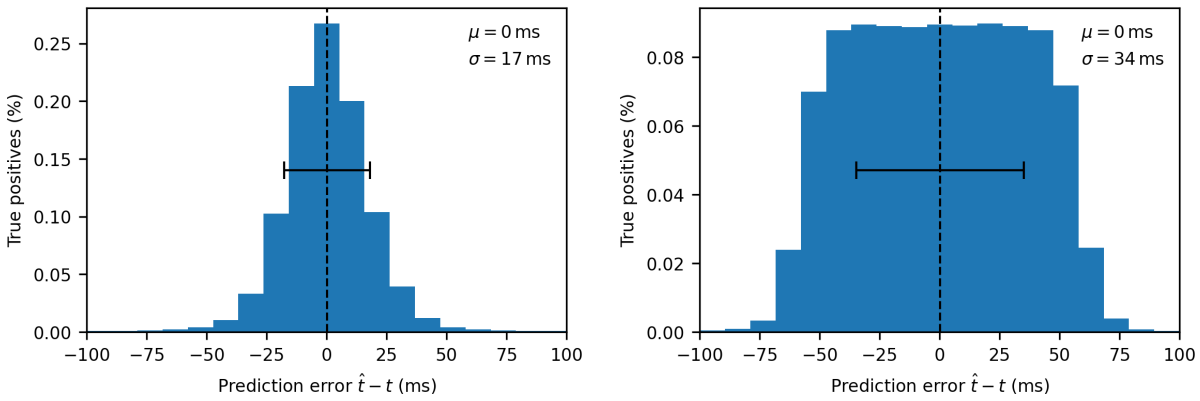


Figure 9.8: Prediction time error $\hat{t} - t$ for DETR (*left*) and the transformer heatmap-regressor (*right*). A vertical line shows the distribution mean μ and an error bar the standard deviation σ. The x-axis limits correspond to the matching criterion $|t - \hat{t}| \leq 0.1$ s.

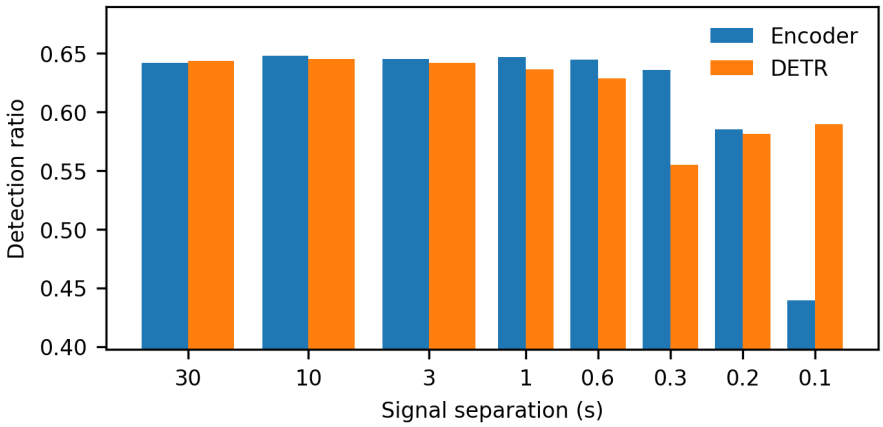


Figure 9.9: Detection ratio as a function of signal separation for the transformer-encoder heatmap regressor (*blue*) and for DETR (*orange*). Computed over 66 k samples with 131 k signals. Uncertainties are too small to be visualized.

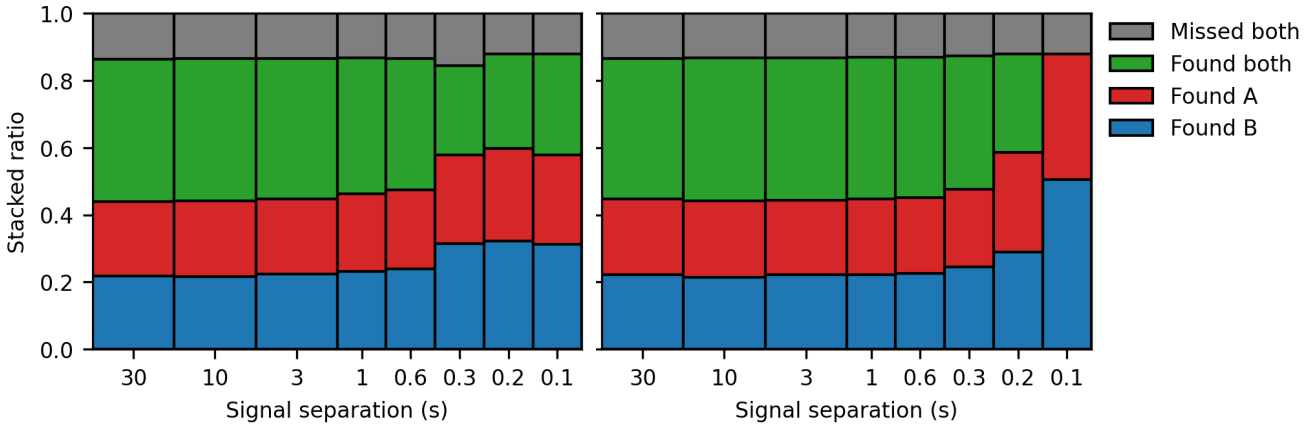


Figure 9.10: Performance with signal separation for DETR (*left*) and for the heatmap regressor (*right*).

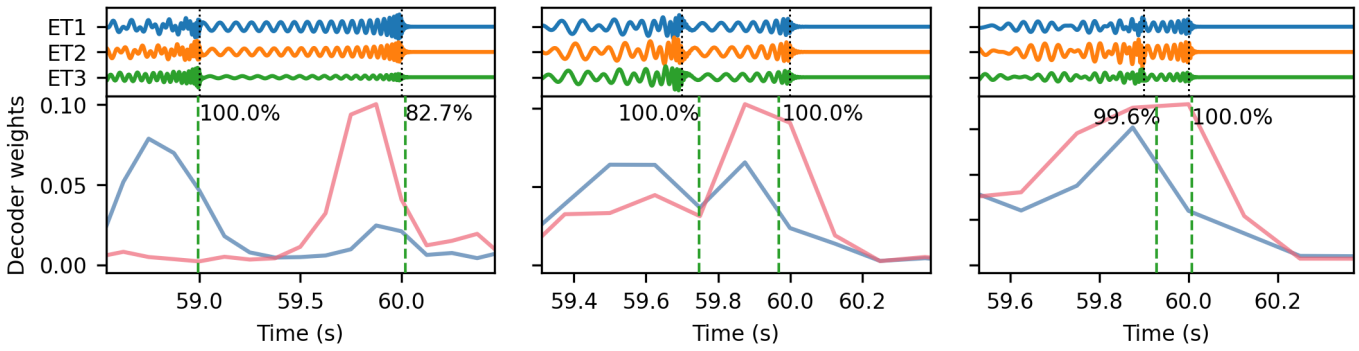


Figure 9.11: Three example DETR outputs for $\Delta t_{\text{coa}} = 1\,\text{s}$ (*left*), $\Delta t_{\text{coa}} = 0.3\,\text{s}$ (*middle*) and $\Delta t_{\text{coa}} = 0.1\,\text{s}$ (*right*).

9.5 Discussion

DETR demonstrates performance comparable to the heatmap transformer while eliminating its intrinsic bottleneck, the heatmap time-resolution. By directly predicting coalescence times, DETR achieves improved time-accuracy. DETR is easily adapted to predict additional signal parameters in future work.

Despite these advantages, the model exhibits reduced performance for low FARs and moderately separated signals with $1\,\mathrm{s} \geq \Delta t_{\mathrm{coa}} \geq 0.3\,\mathrm{s}$. In the regime $\Delta t_{\mathrm{coa}} \leq 0.2\,\mathrm{s}$, performance is comparable to or exceeds that of the heatmap regressor, as DETR is free to predict GW candidates regardless of signal separation.

One possible explanation is that these limitations arise from insufficient model optimization rather than architectural flaws. DETR comprises 6.2 M trainable parameters compared to roughly 2.2 M for the heatmap transformer, while being trained on the same 1 M sample dataset. The learning curves (Figs. 9.3 and 9.4) show that DETR continues improving until epoch 93, considerably later than the heatmap transformer, which overfit at the 16th epoch. This behavior suggests that additional training and larger datasets might further improve performance.

Another potential direction is to better align the matching algorithm DETR uses for training (Hungarian matching) with the one employed for evaluation (greedy matching). It can be argued that greedy matching is more robust in low-FAR operating regimes, where DETR has shown worse performance compared to heatmap regression. This is because at low FARs, greedy matching neglects all predictions with scores $\lesssim 100\,\%$ and assigns only the highest-score predictions. In contrast, Hungarian matching considers *all* predictions for assignment by minimization of some cost function instead. In future work, a one-to-one matching algorithm that is more robust at low FARs like greedy matching could be used for fine-tuning DETR for improved performance in this operating regime.

10 Conclusion

This thesis successfully developed machine-learning models that can detect long-duration, overlapping signals in synthetic ET data. This work focuses on black-hole CBCs and uses 64 s long samples. Each sample contains up to six signals, motivated by the expected event rate of $10^5\,\mathrm{yr}^{-1}$ to $10^6\,\mathrm{yr}^{-1}$ [27]. For one sample, the models output multiple GW candidates, each with a confidence measure and the signal coalescence-time.

In particular, two architectures have been implemented, both showing promising results. These consist of a CNN backbone, as well as transformer components with global attention. A CNN backbone efficiently processes the large input to a more condensed latent representation. The backbone treats each ET detector stream as a different channel and employs heavy downsampling afterwards. Using transformer components, the models can efficiently process the whole latent representation. Exploiting *global* information across the whole time series has shown increased performance for detecting multiple, overlapping signals per sample, compared to models with a *local* receptive field.

The two models differ mostly in their output. One is a transformer heatmap regressor, which outputs a "probability time series". From this, GW candidates are extracted via peak extraction. The other model is the Detection Transformer (DETR) and outputs candidate events directly.

Both models can retrieve a non-zero fraction of signals at low FARs of 1/month, allowing them to make adequately confident detections for low-latency searches. Exact detection ratios are not reported in this conclusion, as these are dataset-dependent, especially on the chosen SNR range, which was not chosen to be realistic here. The results should be understood as proof-of-concept.

These machine-learning algorithms are generally far more computationally efficient than matched filtering. Although not tested, it is reasonable to assume that the models here can process one sample in < 64 s, making them compatible with low-latency searches.

Although both models' performance is roughly comparable, and the heatmap regressor even achieves higher detection ratios for low FARs of 1/month, DETR appears more promising as a basis for future work. In particular, DETR...

- Removes the need for post-processing steps to construct GW candidates.
- Makes more accurate time predictions with a standard deviation of 17 ms compared to 34 ms.
- Can be easily adapted to predict other signal parameters as well. This is important for low-latency searches. Its improved time-prediction accuracy by direct prediction is promising in this case.
- Can make predictions regardless of signal separation. This is not the case for the heatmap regressor and for standard matched filtering searches [31], which only retrieve one of two signals if separations are small.
- Might improve with more data and longer training, as suggested by the learning curves.

Future work should focus on training DETR with more data for more epochs. For increased performance at low FARs, fine-tuning with a reworked Hungarian matching scheme could be investigated. DETR is easily adapted to predict signal parameters as well. This should be studied, focusing on the intrinsic parameters such as the masses and spins first.

Finally, models need to be evaluated on more realistic ET-like data. This includes the simulation of a proper detector stream, which is split into 64 s samples later. These will contain the inspiral parts of signals with coalescence times outside the sample window. Further, instead of sampling the number of signals uniformly, these should follow a Poissonian distribution with the expected ET event rate. Instead of sampling the SNRs uniformly, signals should be simulated ensuring some distance distribution. Usually, this is a second-order power law.

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] S Hild et al. "Sensitivity studies for third-generation gravitational wave observatories". In: *Classical and Quantum Gravity* 28.9 (2011), p. 094013. DOI: 10.1088/0264-9381/28/9/094013.

[6] 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.

[7] Samantha A Usman et al. "The PyCBC search for gravitational waves from compact binary coalescence". In: *Classical and Quantum Gravity* 33.21 (Oct. 2016), p. 215004. ISSN: 1361-6382. DOI: 10.1088/0264-9381/33/21/215004.

[8] Daniel George and E.A. Huerta. "Deep Learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data". In: *Physics Letters B* 778 (Mar. 2018), 64–70. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2017.12.053.

[9] 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.

[10] Lucia Papalini et al. "Can Transformers help us perform parameter estimation of overlapping signals in gravitational wave detectors?" In: *Classical and Quantum Gravity* 42.18 (2025), p. 185012. DOI: 10.1088/1361-6382/adfd33.

[11] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[12] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments*. Oxford University Press, Oct. 2007. ISBN: 9780198570745. DOI: 10.1093/acprof:oso/9780198570745.001.0001.

[13] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Addison-Wesley, 2004.

[14] F Acernese et al. "VIRGO: a large interferometer for gravitational wave detection started its first scientific run". In: *Journal of Physics: Conference Series* 120.3 (2008), p. 032007. DOI: 10.1088/1742-6596/120/3/032007.

[15] F Acernese et al. "Advanced Virgo: a second-generation interferometric gravitational wave detector". In: *Classical and Quantum Gravity* 32.2 (Dec. 2014), p. 024001. ISSN: 1361-6382. DOI: 10.1088/0264-9381/32/2/024001.

[16] T Akutsu et al. "Construction of KAGRA: an underground gravitational-wave observatory". In: *Progress of Theoretical and Experimental Physics* 2018.1 (Jan. 2018), 013F01. ISSN: 2050-3911. DOI: 10.1093/ptep/ptx180.

[17] 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.

[18] 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.

[19] R. Abbott et al. "GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run". In: *Phys. Rev. X* 13 (4 2023), p. 041039. DOI: 10.1103/PhysRevX.13.041039.

[20] A. G. Abac et al. "GWTC-4.0: Updating the Gravitational-wave Transient Catalog with Observations from the First Part of the Fourth LIGO–Virgo–KAGRA Observing Run". In: *The Astrophysical Journal Letters* 1004.2 (2026), p. L22. DOI: 10.3847/2041-8213/ae2c74.

[21] The Nobel Prize. *The Nobel Prize in Physics 2017*. 2017. URL: https://www.nobelprize.org/prizes/physics/2017/summary/ (visited on July 27, 2026).

[22] B. P. Abbott et al. "GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral". In: *Physical Review Letters* 119.16 (Oct. 2017). ISSN: 1079-7114. DOI: 10.1103/physrevlett.119.161101.

[23] ET steering committee. *Design Report Update 2020*. Tech. rep. ET-0007C-20. 2020. URL: https://apps.et-gw.eu/tds/ql/?c=15418.

[24] Marica Branchesi et al. "Science with the Einstein Telescope: a comparison of different designs". In: *Journal of Cosmology and Astroparticle Physics* 2023.07 (2023), p. 068. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2023/07/068.

[25] Samuel Rowlinson et al. "Feasibility study of beam-expanding telescopes in the interferometer arms for the Einstein Telescope". In: *Physical Review D* 103.2 (Jan. 2021). ISSN: 2470-0029. DOI: 10.1103/physrevd.103.023004.

[26] Einstein Telescope. *Candidate locations*. 2026. URL: https://einsteintelescope.eu/about-et/candidate-locations/ (visited on July 27, 2026).

[27] Adrian Abac et al. “The Science of the Einstein Telescope”. In: *Journal of Cosmology and Astroparticle Physics* 2026.03 (2026), p. 081. DOI: 10.1088/1475-7516/2026/03/081.

[28] B. P. Abbott et al. “Multi-messenger Observations of a Binary Neutron Star Merger”. In: *The Astrophysical Journal Letters* 848.2 (Oct. 2017), p. L12. ISSN: 2041-8213. DOI: 10.3847/2041-8213/aa91c9.

[29] Alexander H. Nitz et al. “Rapid detection of gravitational waves from compact binary mergers with PyCBC Live”. In: *Physical Review D* 98.2 (2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.98.024050.

[30] Alex Nitz et al. *gwastro/pycbc: v2.4.0 release of PyCBC*. Version v2.4.0. Oct. 2023. DOI: 10.5281/zenodo.10013996.

[31] Philip Relton et al. “Addressing the challenges of detecting time-overlapping compact binary coalescences”. In: *Physical Review D* 106.10 (Nov. 2022). ISSN: 2470-0029. DOI: 10.1103/physrevd.106.104045.

[32] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[33] Stanford University. *CS231n: Deep Learning for Computer Vision*. URL: https://cs231n.stanford.edu/ (visited on May 5, 2026).

[34] S. Raschka, Y. Liu, and V Mirjalili. *Machine Learning with PyTorch and Scikit-Learn*. 1st ed. Packt Publishing Ltd., Feb. 2022, p. 774. ISBN: 9781801819312.

[35] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd. O’Reilly Media, Inc., 2019. ISBN: 1492032646.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[37] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: https://arxiv.org/abs/1409.0473.

[38] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. 2023. arXiv: 1606.08415 [cs.LG]. URL: https://arxiv.org/abs/1606.08415.

[39] 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.

[40] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML]. URL: https://arxiv.org/abs/1607.06450.

[41] Sebastian Wein. *gw-simulation*. GitHub repository. 2026. URL: https://github.com/sebastianwein/gw-simulation.

[42] Stefan Danilishin and Teng Zhang. *Einstein Telescope sensitivity curves used for CoBA Science study*. Tech. rep. ET-0304B-22. Instrument Science Board, 2023. URL: https://apps.et-gw.eu/tds/ql/?c=16492.

[43] B P Abbott et al. "A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals". In: *Classical and Quantum Gravity* 37.5 (2020), p. 055002. DOI: 10.1088/1361-6382/ab685e.

[44] Kamiel Janssens et al. "Correlated 0.01–40 Hz seismic and Newtonian noise and its impact on future gravitational-wave detectors". In: *Phys. Rev. D* 109 (10 2024), p. 102002. DOI: 10.1103/PhysRevD.109.102002.

[45] Francesco Cireddu et al. "Likelihood for a network of gravitational-wave detectors with correlated noise". In: *Phys. Rev. D* 110 (10 2024), p. 104060. DOI: 10.1103/PhysRevD.110.104060.

[46] Vijay Varma et al. "Gravitational-wave observations of binary black holes: Effect of nonquadrupole modes". In: *Phys. Rev. D* 90 (12 2014), p. 124004. DOI: 10.1103/PhysRevD.90.124004.

[47] 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.

[48] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration. *LVK Algorithm Library - LALSuite*. Free software (GPL). 2018. DOI: 10.7935/GT1W-FZ16.

[49] Duncan Meacher. *Einstein Telescope pre-site selection detector locations technical document*. Tech. rep. T1400308. LIGO Document Control Center, 2014. URL: https://dcc.ligo.org/LIGO-T1400308.

[50] Alexander H. Nitz et al. "Detecting Binary Compact-object Mergers with Gravitational Waves: Understanding and Improving the Sensitivity of the PyCBC Search". In: *The Astrophysical Journal* 849.2 (Nov. 2017), p. 118. ISSN: 1538-4357. DOI: 10.3847/1538-4357/aa8f50.

[51] Sebastian Wein. *ml-models*. GitHub repository. 2026. URL: https://github.com/sebastianwein/ml-models.

[52] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '16. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[53] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html.

[54] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[55] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[56] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *International Conference on Learning Representations (ICLR)*. 2017.

[57] Marlin B. Schäfer et al. "First machine learning gravitational-wave search mock data challenge". In: *Physical Review D* 107.2 (Jan. 2023). ISSN: 2470-0029. DOI: 10.1103/physrevd.107.023021.

[58] Jonathan Tompson et al. "Joint training of a convolutional network and a graphical model for human pose estimation". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'14. Montreal, Canada: MIT Press, 2014, 1799–1807.

[59] Chen Wang et al. "Low-resolution human pose estimation". In: *Pattern Recognition* 126 (2022), p. 108579. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2022.108579.

[60] Hei Law and Jia Deng. "CornerNet: Detecting Objects as Paired Keypoints". In: *International Journal of Computer Vision* 128.3 (2020), pp. 642–656. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01204-1.

[61] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[62] Adrian Bulat, Enrique Sanchez, and Georgios Tzimiropoulos. *Subpixel Heatmap Regression for Facial Landmark Localization*. 2021. arXiv: 2111.02360 [cs.CV]. URL: https://arxiv.org/abs/2111.02360.

[63] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[64] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 213–229. ISBN: 978-3-030-58452-8.

A Quadrupole Moment

In the transverse-traceless gauge, the quadrupole moment tensor takes the form,

$$Q_{ij} = \int \mathrm{d}^3\vec{r}\,(3r_i r_j - \delta_{ij} r^2) T_{00}(t, \vec{r}). \tag{A.1}$$

By expanding the energy density T_{00} as

$$T_{00}(t, \vec{r}) = \sum_l \sum_{m=-l}^{l} \rho_{lm}(t, r) Y_l^m(\theta, \phi) \tag{A.2}$$

with the spherical harmonic functions Y_l^m and coefficients ρ_{lm}, it can be shown that only the $l = 2$ coefficients contribute to the quadrupole moment tensor. This follows from the fact that $3r_i r_j - \delta_{ij} r^2$ can be expanded purely in $l = 2$ spherical harmonics. That is,

$$3r_i r_j - \delta_{ij} r^2 = r^2 \sum_{m=-2}^{2} c_m Y_{l=2}^m, \tag{A.3}$$

which will be verified later. Then,

$$Q_{ij} = \sum_{m'} \sum_{l,m} \tilde{c}_{m'} \int \mathrm{d}r\, r^4 \rho_{lm}(t, r) \int \mathrm{d}\Omega\, (Y_{l'=2}^{m'})^* Y_l^m, \tag{A.4}$$

making use of $(Y_l^m)^* = (-1)^m Y_l^{-m}$, and defining $\tilde{c}_m = (-1)^{-m} c_{-m}$. The orthonormality of the spherical harmonics,

$$\int \mathrm{d}\Omega\, (Y_{l'}^{m'})^* Y_l^m = \delta_{ll'} \delta_{mm'}, \tag{A.5}$$

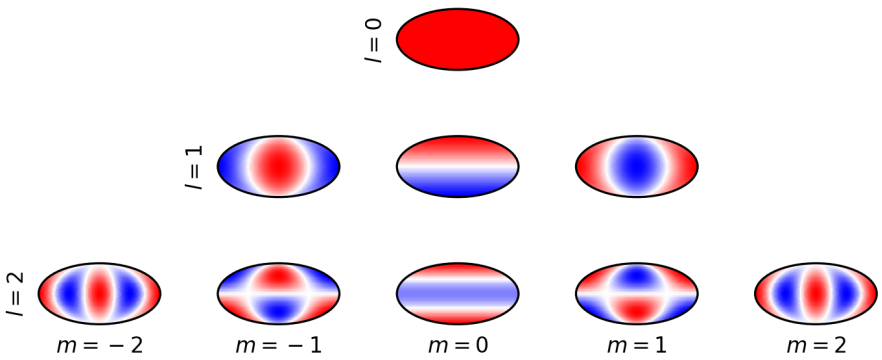


Figure A.1: Visualization of the spherical harmonics up to $l = 2$. The plots show the normalization of $\mathrm{Re}\, Y_l^m(\theta, \phi)$ on a Mollweide projection.

yields

$$Q_{ij} = \sum_m \tilde{c}_m \int \mathrm{d}r\, r^4 \rho_{l=2,m}(t,r). \tag{A.6}$$

This shows that the quadrupole moment tensor is non-vanishing only if the mass distribution has a non-zero $l = 2$ spherical harmonic contribution. From this it is obvious that spherically symmetric ($l = 0$) systems do not radiate. See Fig. A.1 for a visualization of the spherical harmonics. The $l = 2$ distributions are peanut-shaped. Prime examples of time-varying systems with $l = 2$ mass distributions are orbiting binary systems.

Eq. A.3 can be written out explicitly as,

$$3x^2 - r^2 = 3\sqrt{\frac{2\pi}{15}} r^2 (Y_2^2 + Y_2^{-2}) - 2\sqrt{\frac{\pi}{5}} r^2 Y_2^0, \tag{A.7}$$

$$3y^2 - r^2 = -3\sqrt{\frac{2\pi}{15}} r^2 (Y_2^2 + Y_2^{-2}) - 2\sqrt{\frac{\pi}{5}} r^2 Y_2^0, \tag{A.8}$$

$$3z^2 - r^2 = 4\sqrt{\frac{\pi}{5}} r^2 Y_2^0, \tag{A.9}$$

$$xy = -i\sqrt{\frac{2\pi}{15}} r^2 (Y_2^2 - Y_2^{-2}), \tag{A.10}$$

$$yz = i\sqrt{\frac{2\pi}{15}} r^2 (Y_2^1 + Y_2^{-1}), \tag{A.11}$$

$$zx = -\sqrt{\frac{2\pi}{15}} r^2 (Y_2^1 - Y_2^{-1}). \tag{A.12}$$

This can be verified from the $l = 2$ harmonics:

$$Y_2^{-2}(\theta,\varphi) = \frac{1}{4}\sqrt{\frac{15}{2\pi}} \mathrm{e}^{-2i\varphi} \sin^2\theta \quad = \frac{1}{4}\sqrt{\frac{15}{2\pi}} \frac{(x-iy)^2}{r^2}, \tag{A.13}$$

$$Y_2^{-1}(\theta,\varphi) = \frac{1}{2}\sqrt{\frac{15}{2\pi}} \mathrm{e}^{-i\varphi} \sin\theta\cos\theta \; = \frac{1}{2}\sqrt{\frac{15}{2\pi}} \frac{(x-iy)z}{r^2}, \tag{A.14}$$

$$Y_2^{0}(\theta,\varphi) = \frac{1}{4}\sqrt{\frac{5}{\pi}} (3\cos^2\theta - 1) \quad = \frac{1}{4}\sqrt{\frac{5}{\pi}} \frac{(3z^2 - r^2)}{r^2}, \tag{A.15}$$

$$Y_2^{1}(\theta,\varphi) = -\frac{1}{2}\sqrt{\frac{15}{2\pi}} \mathrm{e}^{i\varphi} \sin\theta\cos\theta = -\frac{1}{2}\sqrt{\frac{15}{2\pi}} \frac{(x+iy)z}{r^2}, \tag{A.16}$$

$$Y_2^{2}(\theta,\varphi) = \frac{1}{4}\sqrt{\frac{15}{2\pi}} \mathrm{e}^{2i\varphi} \sin^2\theta \quad = \frac{1}{4}\sqrt{\frac{15}{2\pi}} \frac{(x+iy)^2}{r^2}. \tag{A.17}$$