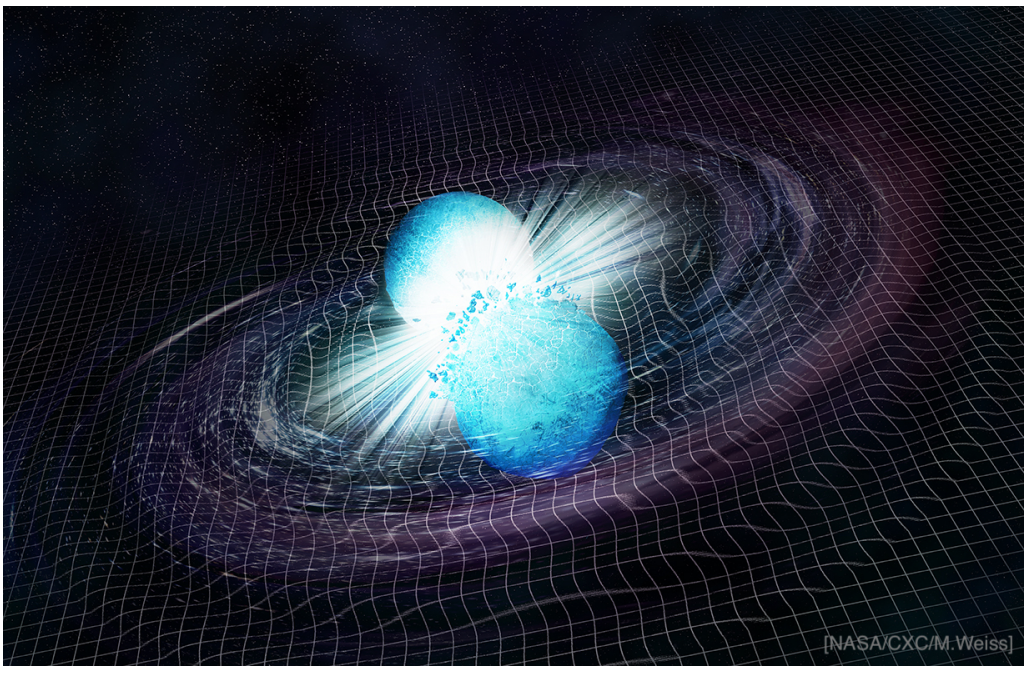
Universiteit Utrecht

Master Experimental Physics

Pre-merger sky localization and parameter estimation of binary neutron star inspirals using normalizing flows

Master Thesis

Wouter van Straalen



Project Supervisor: Prof. Dr. Chris Van Den Broeck [1,3]

Supervisors: Dr. Justin Janquart[1,3], Alex Kolmus[2],

1. Institute for Gravitational and Subatomic Physics (GRASP), Utrecht University
2. Institute for Computing and Information Sciences, Radboud University Nijmegen
3. National Institute for Subatomic Physics (Nikhef), 1098 XG Amsterdam, The Netherlands

June 27, 2024

Abstract

In the coming years, the LIGO and Virgo gravitational wave interferometers are planned to undergo a number of detector upgrades which will improve the detector sensitivity, and together with the construction of next-generation detectors like Einstein Telescope and Cosmic Explorer this should lead to more frequent and louder detections of binary neutron star inspirals. To increase the information gained from these observations, we want to study the inspiral and merger directly with electromagnetic telescopes, which requires us to detect and localize these signals before their merger. Current state-of-the-art localization algorithms rely on matched filtering frameworks, which can introduce biases and only provide point estimates of intrinsic parameters. In this work, we present a normalizing flows based framework that can provide pre-merger sky location, in addition to estimating other parameters relevant for follow-up study, such as the component masses, luminosity distance and inclination angle. We train networks for different maximum frequencies, corresponding to a different time-to-merger. The parameter estimations are better constrained when a larger part of the signal is observed and when the signal is louder, but the networks can also produce well-constrained localizations for smaller parts and quieter signals. The sky localizations produced by the networks are regularly accurate enough to enable follow-up studies.

Contents

1 Introduction

Gravitational waves (GWs) are perturbations in the fabric of spacetime. They were first theoretically described by Albert Einstein in 1916 [1], as a consequence of his famous work on General Relativity [2]. GWs provide an additional method of observing the cosmos; until then we were only able to see the universe using electro-magnetic (EM) emissions, but GWs would allow us to observe the continuous GW emission of asymmetrical neutron stars, study the stochastic GW background, GW bursts emitted by supernovae and the GW chirps emitted by the inspiral of super dense stellar objects [3]. The first indirect observation of GWs was done by Russel Hulse and Joseph Taylor, when they studied the orbital period of a binary pulsar system, and found the orbit decaying due to losing energy to gravitational radiation [4]. Since then, the field of GW science has gained more and more traction with the first direct observation of GWs, named GW150914[1] [5] by the LIGO [6] GW interferometers. Currently, the LIGO and Virgo [7] detector network has made a total of 90 GW detections, with 85 binary black holes (BBHs), 3 black hole neutron star (BHNS) and 2 binary neutron stars (BNS) detections [8]. With planned detector upgrades and upcoming construction of Einstein Telescope [9] and Cosmic Explorer [10], the detector sensitivity will increase significantly.

One of the most interesting detections by the LIGO and Virgo detectors is the first confirmed detection of a BNS inspiral, GW170817 [11]. Later searches revealed that the Fermi Gamma-Ray-Burst Monitor [12] and the INTEGRAL satellites [13] had observed a gamma ray burst (GRB) coming from the same location [14] about two seconds later. This was followed up by multiple EM telescopes, and the aftermath of the merger was observed by 70 observatories across the EM spectrum from radio to X-ray wavelengths. The detection opened up the field of multi-messenger astronomy (MMA), which can combine observations of high-energy neutrinos, ultra high energy cosmic rays, gamma rays, other EM channels and GW data from a single source [15]. The GW170817 detection lead to a lot of discoveries, including but not limited to: an independent measurement of the Hubble constant [16], further constraints on the neutron star equation of state [17], and a comparison between the speed of light and speed of gravity [18].

While GW170817 lead to a plethora of discoveries, we could do even better by directly observing other parts of the inspiral in the EM band. If we can observe and locate the source of a BNS signal *before* the merger, we could observe part of the inspiral and merger directly with EM telescopes. This would provide several interesting opportunities; for example to study the pre-merger magnetosphere interactions between the neutron stars [19], further investigations of the r-process nucleosynthesis which produces heavier elements [20], and observing the X-ray emissions at the merger to determine the state of the remnant object [21].

Several studies have already shown the ability to provide early warnings for BNS inspirals using the LIGO-Virgo detector network at design sensitivity. Some rely on the matched filtering technique, which is partly used to claim GW detections [22–24]. Others rely on machine learning (ML) techniques like convolutional neural networks to produce triggers [25, 26]. These triggers would allow for follow-up investigations to estimate the sky location.

[1] GW observations get named with the date of discovery, so GW150914 corresponds to a GW observation on the 14th of September 2015.

The biggest hurdle to provide pre-merger sky localizations is the stringent speed requirements: the sky location needs to be estimated as fast as possible to ensure the follow-up observations have enough time to find the source before the signal ends. These requirements disqualify classical parameter estimation methods like Markov chain Monte-Carlo (MCMC) [27] and nested sampling [28], since they require a large amount of time to estimate the sky location. Therefore, frameworks specialized in rapid sky localization like `BAYESTAR` [29] and `GWSkyLocator` [30] use alternative methods to find the sky location. The first uses the matched filtering output and marginalized Bayesian parameter estimation to estimate the sky location and luminosity distance. The second also relies on the matched filtering output, but uses a neural network-based framework to provide sky localizations. While efficient, both of these frameworks rely on the matched filtering pipeline to produce results. However, the matched filtering pipeline uses a template bank which is discretely populated with template waveforms. Thus, these can only give a point estimation of the intrinsic parameters of the waveforms which can introduce biases.

ML applications have recently gained increased prominence in GW science. Specifically in GW parameter estimation, multiple studies have used normalizing flow (NF) based frameworks [31–35], because they are fast compared to classical methods with similar accuracy [34].

In this work, we develop a NF based framework capable of rapidly inferring the sky location and other parameters relevant for EM follow-up observations, such as the component masses, luminosity distance and inclination angle, using a pre-merger BNS inspiral. The framework does not require information from other pipelines such as matched filtering based ones, being able to produce sky estimations using only the detector strains. Additionally, because it is an ML-based approach, the majority of the computational cost is up-front, resulting in sub-second parameter estimation.

This work is structured as follows. Sec. 2 introduces the reader to the theory behind GWs, waveform models, the detection of GWs and neutron star multi messenger astronomy. In Sec. 3, we introduce the ML framework used to analyse the BNS inspiral signals. We discuss the implementation and operation of the network in Sec. 4. In Sec. 5 we present the results obtained from our investigations in the accuracy of the framework. Finally, in Sec. 6 we summarize and discuss the results and discuss possible applications.

2 Gravitational waves

2.1 Introduction to gravitational waves

We start the theoretical discussion in this thesis the with a short introduction to General Relativity (GR), with a focus on linearized gravity in order to jump straight to GW theory. In this section, we roughly follow the derivation of Antelis et al. [36] and Chris Van Den Broeck [37].

2.1.1 Einstein equations

In general relativity, a 'gravitational field' is described by a spacetime metric. A metric describes how particles in a gravitational field move around spacetime. For example, the Minkowksi metric, which describes empty space, is given by $\eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$. In GR this is called a 'flat spacetime'. From a metric, the behaviour of particles in the spacetime can be calculated. The relation between the spacetime metric and matter is given by the Einstein field equations [2],

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}, \tag{1}$$

with $G_{\mu\nu}$ the Einstein tensor, G the gravitational constant, c the speed of light and $T_{\mu\nu}$ the energy-momentum tensor. In this equation, $G_{\mu\nu}$ depends on the spacetime metric and its first and second derivatives. Because of the complex dependency on the metric and its derivatives, the calculation the spacetime metric is only analytically possible with a few simple matter distributions. However, we can do a simple approximation to arrive at an expression that describes the existence of GWs.

2.1.2 Linearized gravity

To describe GWs, we use an approximation called linearized gravity. Consider small perturbations a on the Minkowski metric labelled by $h_{\mu\nu}$. The spacetime metric can then be expressed as

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad \|h_{\mu\nu}\| \ll 1 \tag{2}$$

This metric can be used to define an invariant line element

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu \tag{3}$$

with ds^2 the invariant *spacetime distance*. In this context, *invariant* means that this distance is the same for each observer in the spacetime. In essence, Eq.(3) tells us that the metric effects the observed spacetime distance between two events. We will return on what this means in section 2.3.1, when we discuss the affects of GWs on matter.

If we insert the expression for $g_{\mu\nu}$ from Eq.(2) in Eq.(1) and impose the Lorentz gauge invariance of the metric to eliminate some derivatives, we arrive at the linear Einstein field equations[2] given by

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

[2]Follow [37] for a more in-depth derivation.

with $\Box$ the d'Alambertian operator and $\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h^{\alpha}_{\alpha}$. If we look at this equation in vacuum ($T_{\mu\nu} = 0$), it reduces to

$$\begin{aligned} \Box\bar{h}_{\mu\nu} &= 0, \\ \left(-\frac{\partial^2}{c^2\partial t^2} + \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\bar{h}_{\mu\nu} &= 0. \end{aligned} \tag{5}$$

This equation allows for wave solutions for $\bar{h}_{\mu\nu}$. These waves are perturbations of the space-time metric, and we call them gravitational waves. This equation can be solved by a plane wave solution of the form

$$\bar{h}_{\mu\nu} = C_{\mu\nu}\exp(i\kappa_\lambda x^\lambda), \tag{6}$$

with $C_{\mu\nu}$ a symmetric, transverse traceless polarization tensor that contains the amplitudes of the GWs, and κ_λ is the propagation vector. If we assume that our source of GWs is in the xy-plane and the waves are travelling along $\hat{z}$, the polarization tensor reduces to

$$h^{TT}_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\cos(\omega(t - z/c)), \tag{7}$$

with h_+ and $h_\times$ the plus and cross polarization of the gravitational wave, which we explain in more detail later. h^{TT} is called the Transverse-Traceless gauge (TT gauge) because $h^i_i = 0$ and $\kappa^\mu C_{\mu\nu} = 0$, where κ^μ is the propagation vector of the GW.

2.1.3 Multipole expansion

Now that we have established the existence of GWs, we derive how they are created. GWs are emitted by the movement and deformation of matter distributions, so we consider a matter distribution at the source of the GW. The matter distribution and its properties are contained in the energy-momentum tensor, $T_{\mu\nu}$. If we look at Eq.(4) and use a Green's function to express it in integral form, we get

$$\begin{aligned} \bar{h}_{\mu\nu}(t, \mathbf{x}) &= -\frac{16\pi G}{c^4}\int_V d^4\mathbf{x}' G(\mathbf{x} - \mathbf{x}')T_{\mu\nu}(\mathbf{x}'), \\ G(\mathbf{x} - \mathbf{x}') &= -\frac{\delta(t_{\text{ret}} - t')}{4\pi|\mathbf{x} - \mathbf{x}'|}, \end{aligned} \tag{8}$$

with V the volume of the source that generates the GWs and $G(\mathbf{x} - \mathbf{x}')$ a Green's function dependent on the retarded time t_{ret}. The retarded time is the time a GW needs to arrive from a certain position $\mathbf{x}'$ at the observer at position $\mathbf{x}$, defined as $t_{\text{ret}} = t - \frac{|\mathbf{x}-\mathbf{x}'|}{c}$. If we apply the Green's function to the energy-momentum tensor and integrate the time integral, Eq.(8) reduces to

$$\bar{h}_{\mu\nu}(t, \mathbf{x}) = \frac{4G}{c^4}\int_V d^3\mathbf{x}' T_{\mu\nu}(t - |\mathbf{x} - \mathbf{x}'|/c, \mathbf{x}'). \tag{9}$$

With appropriate gauge transformation choices, we can set $\bar{h}_{0\mu} = 0$ outside the source, so we can focus on the spatial components. Far away from the source at a distance r, we can

approximate $|\mathbf{x} - \mathbf{x}'| \approx r$ to simplify the expression. With these approximations, Eq.(9) becomes

$$\bar{h}_{ij}(t, \mathbf{x}) = \frac{1}{r}\frac{4G}{c^4}\int_V d^3\mathbf{x}' T_{ij}(t - r/c, \mathbf{x}'). \tag{10}$$

We can use gauge transformations to convert this to the TT gauge using a linear operator,

$$h^{TT}_{\mu\nu} = \Lambda_{ijkl}(\hat{\mathbf{n}})\bar{h}^{kl}, \tag{11}$$

with $\Lambda_{ijkl}(\hat{\mathbf{n}})$ the operator that converts to the TT gauge and $\hat{\mathbf{n}}$ a unit vector in the direction of propagation[3]. Using $\Lambda_{ijkl}(\hat{\mathbf{n}})$, Eq.(10) can be expressed as

$$h^{TT}_{ij}(t, \mathbf{x}) = \frac{1}{r}\frac{4G}{c^4}\Lambda_{ijkl}(\hat{\mathbf{n}})\int_V d^3\mathbf{x}' T^{kl}(t - r/c, \mathbf{x}'). \tag{12}$$

Now, using $T_{ij} = \delta^k_i\delta^l_j T^{kl} = (\partial^k x_i)(\partial^l x_j)T^{kl}$ and the conservation law $\partial_\mu T^{\mu\nu} = 0$ we can use partial integration to arrive at the expression

$$h^{TT}_{ij}(t, \mathbf{x}) = \frac{1}{r}\frac{2G}{c^4}\Lambda_{ijkl}(\hat{\mathbf{n}})\ddot{M}^{kl}(t - r/c), \tag{13}$$

with $\ddot{M}^{kl}$ the second order time derivative of the mass multipole moment

$$M^{ij} \equiv \frac{1}{c^2}\int d^3\mathbf{x} T^{00} x^i x^j. \tag{14}$$

If we choose $\hat{\mathbf{n}} = \hat{\mathbf{z}}$ and write out the expression for h^{TT}_{ij}, we arrive at the expressions of h_+ and $h_\times$ in terms of the mass multipole moment, given by

$$\begin{aligned} h_+ =& \frac{1}{r}\frac{G}{c^4}\left(\ddot{M}^{11} - \ddot{M}^{22}\right), \\ h_\times =& \frac{2}{r}\frac{G}{c^4}\ddot{M}^{12}. \end{aligned} \tag{15}$$

This approximation looks reasonably compact, but it is only a first order approximation, so it must be taken with a pinch of salt.

2.1.4 Circular orbit

Now that we have an expression for the plus and cross polarizations, we can calculate the GWs originating from two orbiting point particles.

The particles have mass m_1 and m_2, with a distance of $2R$ between them. If we choose the origin to be at the center of mass, we can define a unit vector $\hat{e}(t)$ that points from the origin to the first particle. The positions of the particles can then be expressed as

$$\begin{aligned} \mathbf{x}_1(t) =& \frac{\mu}{m_1}R\hat{e}(t), \\ \mathbf{x}_2(t) =& -\frac{\mu}{m_2}R\hat{e}(t), \end{aligned} \tag{16}$$

[3]The $\Lambda_{ijkl}(\hat{\mathbf{n}})$ operator is constructed from a combination of projection operators $P_{ij}(\hat{\mathbf{n}}) \equiv \delta_{ij} - n_i n_j$.

with $\mu = m_1 m_2/(m_1+m_2)$ the reduced mass, $\hat{e}(t) = (\cos(\omega t), \sin(\omega t)\cos(\iota), \sin(\omega t)\sin(\iota))$. ω is the orbital frequency and ι is the inclination of the orbital plane with respect to a normal vector in the direction of the source[4]. Now we calculate the mass multipole moment of this matter distribution from Eq.(14) and substitute it in Eq.(15) to get the polarizations. The equations we get are

$$\begin{aligned} h_+ &= -\frac{4}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/3}\left(\frac{\omega}{c}\right)^{2/3}\frac{1+\cos(\iota)}{2}\cos(2\omega t_{\text{red}}), \\ h_\times &= -\frac{4}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/3}\left(\frac{\omega}{c}\right)^{2/3}\cos(\iota)\cos(2\omega t_{\text{red}}), \end{aligned} \tag{17}$$

with $\mathcal{M}_c = \frac{(m_1 m_2^{3/5})}{(m_1+m_2)^{1/5}}$ the chirp mass. We used Kepler's law to substitute $R = \left(\frac{GM}{\omega^2}\right)^{1/3}$. These equations describe circular motion, but if two particles on a circular orbit emit GWs, they must also lose orbital energy equal to the energy emitted by the GWs. This is included in the next section.

2.1.5 Quasi-circular inspiral

To describe the inspiral, we need to include the time dependence of the frequency due to the energy loss of emitted gravitational radiation. This energy loss manifests in a reduction in the orbital frequency[5], given by the equation

$$\begin{aligned} \dot{f}_{\text{gw}}(t_{\text{red}}) =& \frac{96}{5}\pi^{8/3}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{5/3} f_{\text{gw}}^{11/3}(t_{\text{red}}), \text{ which gives} \\ f_{\text{gw}}(t) =& \frac{1}{\pi}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{-5/8}\left(\frac{5}{256}\frac{1}{\tau(t)}\right) \end{aligned} \tag{18}$$

with $f_{\text{gw}} = \pi\omega$ and $\tau = t_c - t$, with t_c the time of coalescence. The coalescence time is the time where the frequency diverges. In reality, the inspiral stops earlier around the innermost stable circular orbit (ISCO) distance $R_{\text{ISCO}} \sim 6GM/c^2$, after which the object plunge into each other and the inspiral stops. If we assume a *quasi-circular inspiral*[6], we can include the time dependence in Eq.(17) by substituting $\omega \to \omega(t_{\text{red}}), \omega t_{\text{red}} \to \Phi(t_{\text{red}})$, with Φ the angular velocity. When we include all these expressions and approximations, the equation for the polarisations is given by

$$\begin{aligned} h_+ &= -\frac{4}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/3}\left(\frac{\pi f_{\text{gw}}(t_{\text{red}})}{c}\right)^{2/3}\frac{1+\cos(\iota)}{2}\cos(\Phi_{\text{gw}}(t_{\text{red}})), \\ h_\times &= -\frac{4}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/3}\left(\frac{\pi f_{\text{gw}}(t_{\text{red}})}{c}\right)^{2/3}\cos(\iota)\cos(\Phi_{\text{gw}}(t_{\text{red}})). \end{aligned} \tag{19}$$

with

$$\Phi_{\text{gw}}(t_{\text{red}}) = -2\left(\frac{5G\mathcal{M}_c}{c^3}\right)^{-5/8}\tau^{5/8}(t) + \Phi_c, \tag{20}$$

[4]Fig(5) illustrates the definition of the inclination angle.

[5]See section VIII B from [37].

[6]We assume R to be approximately constant over a single orbit.

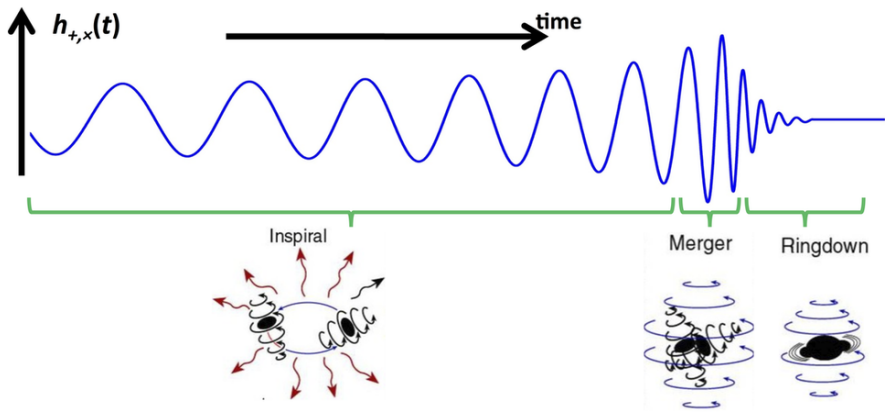


Figure 1: Figure illustrating the different parts of a CBC. Figure taken from [40].

where Φ_c is the phase of coalescence. We now have a first approximation of the GWs emitted by two inspiraling point particles. This is the simplest approximation for a waveform[7], but it includes a lot of approximations that limit its accuracy. In the next section we will not do full derivations, but rather describe the process and idea behind more accurate approximations.

2.2 Waveform models

In GR, there are no analytical solutions to the two body problem. Therefore, if we want to achieve more accuracy for our waveform model, we consider pertubative and numerical methods. We briefly discuss the post-Newtonian (PN) [38] and the self force perturbation theory (SFPT) [39] and their respective domains of applicability, in addition to a numerical method called numerical relativity (NR). Afterwards, we discuss how to combine these approximations into a usable waveform model. In the rest of this section we work in geometrized units, i.e. $c = 1$ and $G = 1$.

2.2.1 Post-Newtonian approximation

From the perspective of waveform modelling, the compact binary coalescence (CBC) of two objects can be divided into three distinct parts: the inspiral, merger and ringdown (see Fig.1). During the inspiral, the separation between the two objects is large with respect to their size. In this part, the dynamics can be modelled using the PN approximation to GR. We assume

$$v \ll 1, \quad \frac{M}{R} \sim (v)^2 \ll 1, \tag{21}$$

meaning that the bodies in the system are moving slowly with respect to the speed of light. Additionally we assume that the gravitational field is 'weak', which is indeed what we expect during the inspiral phase because the objects are still far away from eachother. In the PN

[7]We use the definition for the term waveform for a specific equation for $h_+, h_\times$ depending on the source parameters.

approximation, the GW polarizations have the following general structure [38]

$$h_{+,\times} = \frac{2\mu v^2}{r} \sum_{p\geq 0} v^p H^{(p)}_{+,\times} + \mathcal{O}\left(\frac{1}{r^2}\right) \tag{22}$$

where we redefine the variable v as a frequency related parameter using Kepler's law:

$$v^2 \equiv (M\omega)^{2/3} = \frac{M}{R}\left\{1 + \mathcal{O}\left(\frac{1}{c^2}\right)\right\}. \tag{23}$$

The parameter H in Eq.(22) contains the different expansion coefficients. The leading order term reads

$$H^{(0)}_{+} = -(1+\cos^2(\iota))\cos(\psi), \quad H^{(0)}_{\times} = -2\cos(\iota)\sin(\psi) \tag{24}$$

and has similar terms to Eq.(19), except now we also introduce the "tail-disordered" phase $\psi = \Psi_{\rm gw}(t_{\rm red}) - 6v^3\ln(v)$ where the extra term comes from the scattering of the GW off the static curvature of the binary system. For GW data analysis, having a high precision in the phase is significantly more important than the precision in the amplitude because the latter scales with the luminosity distance when the waveform is projected onto the detector frame. So, commonly only the leading order term in amplitude $H^{(0)}_{+,\times}$ is retained, while the PN corrections to the orbital phase evolution are included. The orbital phase can be expanded in the general structure

$$\Phi(v) = -\frac{1}{32\eta}\frac{1}{v^5}\left\{1 + \mathcal{O}(v^2) + \mathcal{O}(v^3) + \mathcal{O}(v^4) + ...\right\}, \tag{25}$$

with $\eta = (m_1 m_2)/(m_1+m_2)^2$ the symmetric mass ratio. The general structure of the PN expansion can be iterated upon and more terms can be added to give the desired accuracy. How many terms you include in this expansion is called the used PN order. We will not discuss the higher order approximations because it is beyond the scope of this work. We refer the reader to [41] for a more complete derivation with the relevant coefficients.

2.2.2 Self force perturbation theory

Another important part of the puzzle in waveform modelling is black hole perturbation theory [39], which allows for more accurate modelling of small mass ratio systems. In this setup, the 'zeroth-order' system is that of a test particle (the lighter object) moving along a path in a fixed background spacetime of the heavier object. This situation is then expanded by including corrections order by order in the mass ratio. At first order, the gravitational field of the small object is a linear perturbation on top of the background spacetime. This correction gives rise to a gravitational self force which gradually diverts the object from its initial path. Therefore, this approximation is called self force perturbation theory (SFPT). In this approximation, it is the gravitational self force that is responsible for the decaying orbit. The theoretical description of gravitational self force is quite involved, so we refer the interested reader to [39].

2.2.3 Numerical relativity

In some regions of parameter space and some parts of the signal, the approximations mentioned before are no longer valid. The PN and SFPT approximations break down when the objects are very close together during the merger phase as defined in Fig.(1). For these regions, we need a different way to find solutions to the Einstein equations: numerical relativity (NR). Here we give a brief introduction into the basis of NR. For a more concrete formulation we would like to refer the reader to the work of C. Palenzuela [42].

To evolve a numerical solution of a spacetime manifold, we need to calculate the components of the spacetime metric $g_{\mu\nu}$ and evolve them in time. A priori, we have 16 different equations to solve. We can reduce this by using the Bianchi identities from differential geometry, defined by

$$\begin{aligned} \nabla_\mu G^{\mu\nu} =& 0 \rightarrow \nabla_\mu T^{\mu\nu} = 0, \quad \text{with} \\ \nabla_\mu T^{\mu\nu} =& \partial_\mu T^{\mu\nu} + \Gamma^\mu_{\mu\alpha} T^{\alpha\nu} + \Gamma^\nu_{\mu\alpha} T^{\mu\alpha}, \\ \Gamma^\alpha_{\mu\nu} =& \frac{1}{2} g^{\alpha\beta} (\partial_\mu g_{\nu\beta} + \partial_\nu g_{\mu\beta} - \partial_\beta g_{\mu\nu}), \end{aligned} \tag{26}$$

with ∇_μ the covariant derivative[8] and $\Gamma^\alpha_{\mu\nu}$ the Christoffel symbols. By using Eq.(26) we reduce the degrees of freedom to 8; four coordinates and four constraints. These can then be described using the so-called $3+1$ decomposition to split the space and time components. These components can then be integrated using the $3+1$ formulation of the Einstein equation to get the dynamics. For a more in-depth description, we refer the reader to [42].

One of the drawbacks of using NR to describe the spacetime metric is the immense computational cost, requiring supercomputers to complete the simulations. Because of this, the use of NR is constrained to modeling only part of the CBC signal and cannot efficiently be used to calculate full waveforms for multiple source parameters. However, the NR simulations are still useful to waveform modelling by calibrating or fitting different types of models to NR results. Due to this, the accuracy of these simulations is very important for conventional waveform models like the `IMRPhenom` waveforms [43].

2.2.4 Combining different approximations

As mentioned, the discussed approximations hold in different regimes of parameter space and different parts of the signal. The PN approximation focuses on low-velocity bound state systems while the SFPT approximation focuses on low mass systems. Fig.(2) shows a comparison between the parameter regimes of the approximations discussed so far.

In practice, the results from the different theories are usually combined to create a waveform model. One option is the so-called effective one body (EOB) waveform [45]. This waveform is constructed by recasting the two-body problem given by PN theory into simpler one-body dynamics. The waveform obtained extends the domain of validity of the PN approximation and can then be fitted to the NR waveforms [45]. There is a significant downside however: EOB waveforms can be slow to evaluate, and to do parameter estimation one needs to evaluate a lot of waveforms for different parameters, which takes a long amount of

[8] The covariant derivative is a generalization of the partial derivative on a manifold.

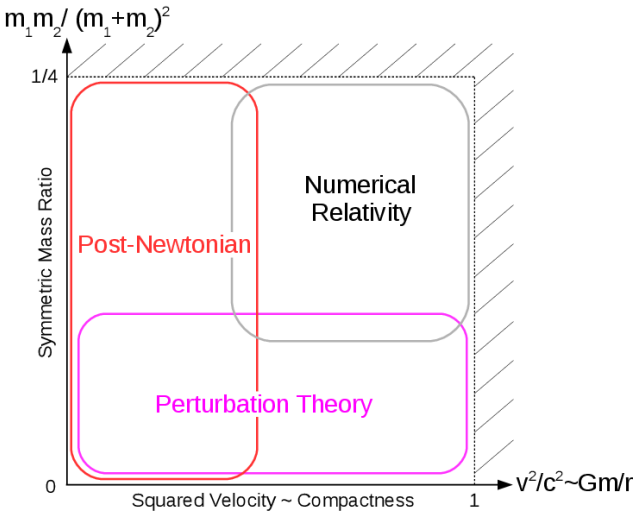


Figure 2: Figure showing the parameter regimes for PN, SFPT and NR theories in the CBC case depending on the symmetric mass ratio and the average velocity. Figure taken from [44].

time. Because of this, we use a different waveform model: the phenomenological inspiral-merger-ringdown (`IMRPhenom`) waveform. The philosophy for these waveform models is to interpolate between the NR and PN simulations by modelling simple fit functions for the frequency domain behaviour of the GWs [46]. The `IMRPhenom` models are faster to evaluate than the EOB models with a comparable precision, which makes them ideal for parameter estimation [46].

2.2.5 `IMRPhenomD` waveform

In the `IMRPhenom` waveform family there are still multiple options to choose from depending on what type of systems one wants to study. In this work, we use a `IMRPhenomD` waveform modified to include tidal deformability. It is constructed out of different parts used to fit to the EOB and NR waveforms. Here we discuss a short description of the constituent parts of the waveform. For a more complete description please refer to [47, 48].

The waveform depends on the intrinsic GW parameters, which are the masses of the objects and their spins. The `IMRPhenomD` waveform assumes that the spins of the object are aligned and non-precessing. To model the waveform, we need to construct a fitting function for the phase and the amplitude in each region: the inspiral, the merger and the intermediate region. The waveform also contains a ringdown region, which is calculated with a perturbation on the resulting object which is then evolved over time. The ringdown region for BNS inspirals is tapered away because the frequency of this part of the waveform is higher than the detectors maximum detectable frequency, so we will not go into details here about this part. The inspiral can be described with a PN waveform called `TaylorF2` [49] which is fitted to the EOB waveform. The phase is given by

$$\Phi_{\text{ins}} = \Phi_{\text{F2}} + \frac{1}{\eta}\left(\alpha_0 + \alpha_1 f + \frac{3}{4}\alpha_2 f^{3/4} + \frac{3}{5}\alpha_3 f^{5/3} + \frac{1}{2}\alpha_4 f^2\right), \tag{27}$$

with $\Phi_{\rm F2}$ the `TaylorF2` phase, $\eta = m_1 m_2/(m_1+m_2)^2$ the symmetric mass ratio and α_i the fitting parameters. The inspiral amplitude is given by

$$A_{\rm ins} = A_{\rm F2} + A_0 \sum_{i=1}^{3} \rho_i f^{\frac{6+i}{3}}, \tag{28}$$

with $A_{\rm PN}$ the amplitude obtained from the PN expansion, and ρ_i fitting parameters accounting for effects not included in the PN formalism. The other parts of the model are obtained by constructing an ansatz on the derivative of the phase to remove ambiguity on the reference phase [47]. We discuss the results of the integrations to gain some intuition on how the waveforms are constructed.

The characteristic feature in the merger and ringdown part of the GW is a dip in the phase derivative. This can be modelled by adding a damping term dependent on the ringdown frequency $f_{\rm RD}$ and its damping frequency $f_{\rm damp}$. This results in the equation

$$\eta\Phi_{\rm MRD} = \beta_0 + \beta_1 f^{-1}\frac{4}{3} + \beta_2 f^1 + \beta_3 f^{3/4} + \beta_4 + \arctan\left(\frac{f-\beta_5 f_{\rm RD}}{f_{\rm damp}}\right) \tag{29}$$

with β_0 a integration constant, β_1 a time shift (which will be determined when we impose smooth transitions between the waveform regions) and the other β parameters are fitting parameters. The amplitude is given by a mixture of a Lorentzian and decreasing exponential:

$$A_{\rm MRD} = A_0\gamma_1 \frac{\gamma_3 f_{\rm damp}}{(f-f_{\rm RD})^2 + (\gamma_3 f_{\rm damp})^2)} \exp(\frac{\gamma_2(f-f_{\rm RD})}{\gamma_3 f_{\rm damp}}) \tag{30}$$

with the γ terms fitting parameters.

The transition region has a dominant phase derivative evolution of f^{-1} with an added corrective term in f^{-4} to account for observed deviations. The phase in the intermediate region is given by

$$\eta\Phi_{\rm int} = \delta_0 + \delta_1 f + \delta_2 \ln(f) - \frac{\delta_3}{3} f^{-3}, \tag{31}$$

with δ_0 an integration term and δ_1 a time shift term. The rest are fitting parameters. The amplitude in this region is represented by a forth order polynomial whose boundary conditions are fixed by requiring matching amplitudes with the merger and the inspiral:

$$A_{\rm int} = A_0(\epsilon_0 + \epsilon_1 f + \epsilon_2 f^2 + \epsilon_3 f^3 + \epsilon_4 f^4), \tag{32}$$

where the ϵ coefficients are fixed with the boundary conditions.

Now that we have the expressions for the phase and amplitude of each part, the full waveform can be constructed by 'stitching' together the expressions for each part. To get waveforms that are valid for binary neutron star (BNS) systems, we also need to include tidal effects due to the gravitational interaction between the two stars.

2.2.6 Tidal waveforms

In comparison to BBH inspirals, the BNS inspirals have some additional complexity which is not included in the waveforms discussed so far. Because neutron stars consist of matter,

the gravitational field of nearby objects leads to deformations, just like the presence of the moon affects the tides of the ocean on earth. These tidal deformations can leave an imprint on the GWs generated by BNS inspirals. In waveform generation, this effect can be included by adding a perturbation on the regular BBH waveform [50]. Since this work focuses on BNS inspirals, the waveform we use includes such a tidal deformability perturbation.

Adding tidal effects to the waveform mainly affects the GW phase. To describe the phase effects of including tidal parameters in the waveform, we work with the phase as a function of the dimensionless GW frequency $\hat{\omega} = M\partial_t\phi(t)$. Then we split the GW phase into three parts:

$$\Phi(\hat{\omega}) = \Phi_0(\hat{\omega}) + \Phi_T(\hat{\omega}) \tag{33}$$

with $\Phi_0(\hat{\omega})$ the phase of the original waveform and $\Phi_T(\hat{\omega})$ the phase contribution due to the tidal effects. The leading-order contribution of the phase reads [51]

$$\Phi_T(\hat{\omega}) = -k^T_{\text{eff}} \frac{c_{\text{Newt}} x^{5/2}}{X_A X_B}(1 + c_1 x) \tag{34}$$

, with $x = (\hat{\omega}/2)^{2/3}$, X_A, X_B parameters related to the spins of the neutron stars, k^T_{eff} the effective tidal coupling constant, $c_{\text{Newt}} = -13/8$, and $c_1 = 1817/364$. A complete description of these parameters falls outside of the scope of this work, so we refer the interested reader to [51] and its cited articles.

In addition to the phase contributions, the `NRTidalv2` perturbation also includes additional tapering in the ringdown part of the signal. Since we do not use this part of the signal in the rest of this work, this is not relevant for us.

In this work, we use `NRTidalv2` developed by Dietrich et al. [52] to implement the tidal deformabilities. As a basis waveform for the `NRTidalv2` waveform, we use `IMRPhenomD`, thus limiting ourselves to aligned spin waveforms. This simplification is sufficient for this proof-of-concept work. We use the waveform implemented in the `Ripple` package [53].

2.3 Detection of gravitational waves

Now that we have discussed the source and modelling of gravitational wave signals, we move on to detecting GWs on Earth by discussing the effects of GWs on matter, the GW interferometers that can be used to detect them and a short introduction on GW parameter estimation.

2.3.1 Gravitational waves and matter

To discuss the effects of a passing GW on matter, we return to the metrics discussed in section 2.1, in particular Eq.(3). We start with a simple example: the interaction of a plane wave solution with matter. We recall the plane wave solution from Eq.(7):

$$h^{TT}_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cos(\omega(t - z/c)) \tag{35}$$

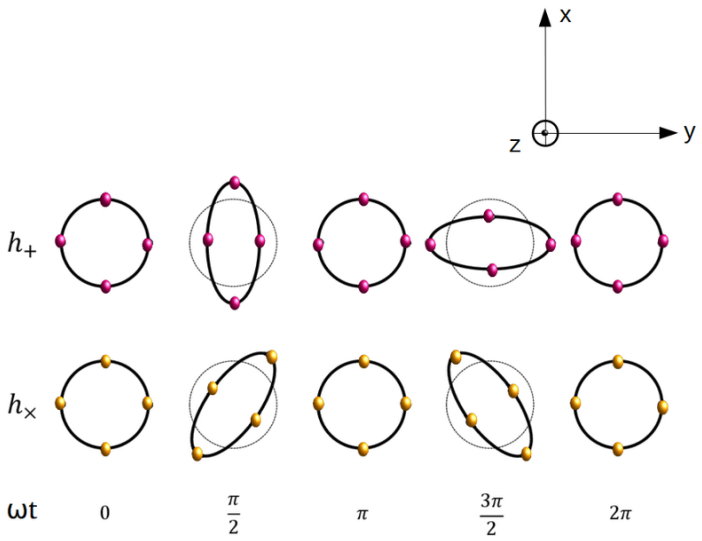


Figure 3: Illustration of the effect of a plane wave coming from the z direction on a ring of particles in the x, y plane. Figure taken from [54]

If we insert this metric into the Eq.(3) we get an expression for the line element:

$$\begin{aligned} ds^2 = &- c^2dt^2 + (1 + h_+ \cos(\omega(t - z/c)))\, dx^2 + (1 - h_+ \cos(\omega(t - z/c)))\, dy^2 \\ &+2h_\times \cos(\omega(t - z/c))dxdy + dz^2. \end{aligned} \tag{36}$$

If we take $h_\times = 0$ and $h_+ \neq 0$ the x and y direction will stretch and squeeze periodically. We can illustrate this by considering this effect on a ring of particles, like in Fig.(3). First the ring of particles will be stretched in the x direction and squeezed in the y direction, and one period later they will be squeezed in the x direction and stretched in the y direction, making a + shape. This effect is illustrated in the first row of Fig.(3). Next we consider the case where $h_+ = 0$ and $h_\times \neq 0$. If we rotate the x, y axis by 45° the deformation will happen in the same way along the rotated x, y axis, therefore it will make a $\times$ shape, as shown in the second row of Fig.(3). So, in conclusion, the effect of a passing GW is that the distance between points contracts and expands periodically. If we want to measure the passing of a GW, we need to measure the distance between two points very exactly.

2.3.2 Gravitational wave interferometers

To measure GWs in practice, we use something called an interferometer. An interferometer is a detector that uses light to measure the difference in travel time between the two detector arms. This is equivalent to measuring the length difference between the two arms by using light to measure the distance. In small scale, interferometers can be used to measure the breaking index of materials and were used in the famous experiment by Michelson and Morley to detect the presence of the 'aether', a hypothetical medium through which electromagnetic radiation travels but its existence was partly disproved by their experiment [55].

The general idea behind an interferometer is quite simple. A laser is sent onto a mirror that splits it between two arms. At the end of the arms, the light is reflected by a mirror and it travels back to a light detector. If the length of the two arms is identical, no light is detected since it interferes destructively at the detector. If there is a difference in the arm length, there will be a phase difference between the different light beams arriving at the detector and thus it will not interfere destructively and the detector produces an output. We call this output the strain h of the detector.

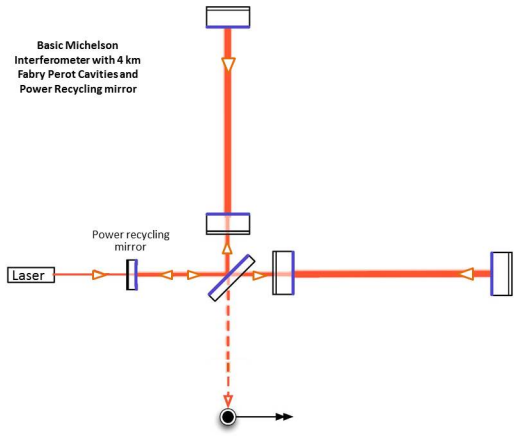
On Earth, there are a few interferometers used to detect GWs. The first GW detector constructed was the GEO600 detector [56] in Germany, to be used as a testbed for further detector development. The GW detections made so far were done with the LIGO Hanford and LIGO Livingston [6] detectors in the US and the Virgo [7] detector in Italy. The KAGRA [57] detector in Japan is undergoing calibration and is expected to join the fifth observing run in 2025. Upcoming next generation detector constructions include Einstein Telescope [9] and Cosmic Explorer [10], and are planned to be operational around 2035. In addition to this, there are also plans to build a space-based GW interferometer called LISA [58], focused on low frequency ($\mathcal{O} \sim 0.1$ mHz $- 1.0$ Hz) detections.

In the rest of this work, we focus on the LIGO and Virgo detectors, since those are the ones that are currently in operation. These detectors differ slightly in construction: the LIGO detectors have a arm length of 4 km, while Virgo has arms with a length of 3 km, with different installed components. Fig.(4a) shows a schematic representation of the interferometer setup.

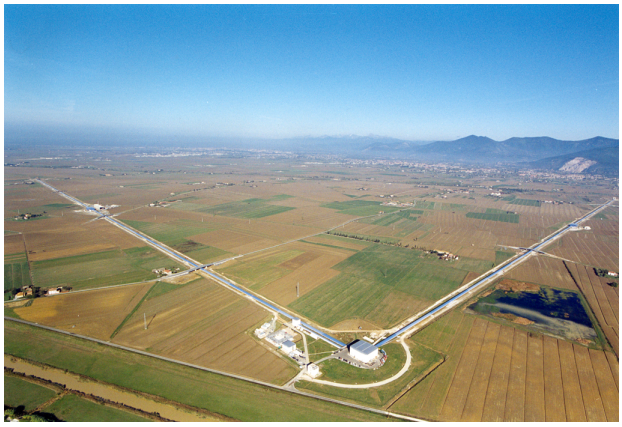
To be able to measure a GW, the interferometers need to measure length differences of $\mathcal{O} \sim 10^{-18}$ meters. For reference, this is about 1000 times smaller then the radius of a proton. To make a instruments that are sensitive enough to measure this, a number of methods are employed[9]:

- The interferometers arms are long: the LIGO detectors have an arm length of 4 km and the Virgo detector has an arm length of 3 km.
- To further increase the light travel time along the detector arms, the detectors also include Fabry-Perot light cavities [59]. These cavities reflect the light in the arms multiple times which increases the light travel time, which increases the effective arm length .
- Seismic vibrations are reduced by using a seismic isolation system and a quadruple suspension system which mount the mirrors and detector components [6].
- The whole apparatus is contained within a ultra-high vacuum chamber to make sure other particles cannot interfere with the laser beam [60].
- To decrease lost laser power from light travelling back to the laser from the beam splitter, a power recycling mirror is used. This mirror fully transmits the light from the laser but reflect the light from the other side back towards the detector [61, 62].
- The sensitivity of the detector is enhanced further by using a signal recycling mirror which can be used to enhance the sensitivity of the detector in a specified bandwidth by

[9]Not every GW detector mentioned has these technologies installed.



(a) Basic Michelson interferometer with power recycling and Fabry-Perot cavities. Figure from [64]



(b) An aerial photo of the Virgo detector in Italy. This interferometer has an arm length of 3 km, which is further extended by light storage arms. Picture from [65].

Figure 4

arranging both the laser light and GW-induced sideband to be resonant in the optical system [63].

For a more complete description of the construction and components of the modern GW detectors, please refer to the aLIGO [6] and Virgo [7] papers.

2.3.3 Detector response

To see how an interferometer responds to a passing GW, we will determine the length difference between the two arms, which corresponds to the strain it measures. Consider a interferometer in the origin with arms of length L along the x and y axis. A plane GW coming from the z axis passes the detector. To measure the length of the detector arms, we can use the equation

$$\mathcal{L} = \int_0^L \sqrt{ds^2}, \tag{37}$$

with ds^2 given by Eq.(3). We want to calculate the physical length of the x arm, so $dt = dy = dz = 0$. We define $h_+(t) = h_+ \cos(\omega t)$. This results in

$$\mathcal{L}_x = \int_0^L \sqrt{1 + h_+(t)} dx = L\sqrt{1 + h_+(t)} \approx L(1 + \frac{h_+ \cos(\omega t)}{2}) \tag{38}$$

Now we can calculate the length difference,

$$\delta L_x = \mathcal{L}_x - L = h_+(t)\frac{L}{2}. \tag{39}$$

We get a similar expression if we instead consider the lenght difference along the y arm:

$$\delta L_y = -h_+(t)\frac{L}{2}. \tag{40}$$

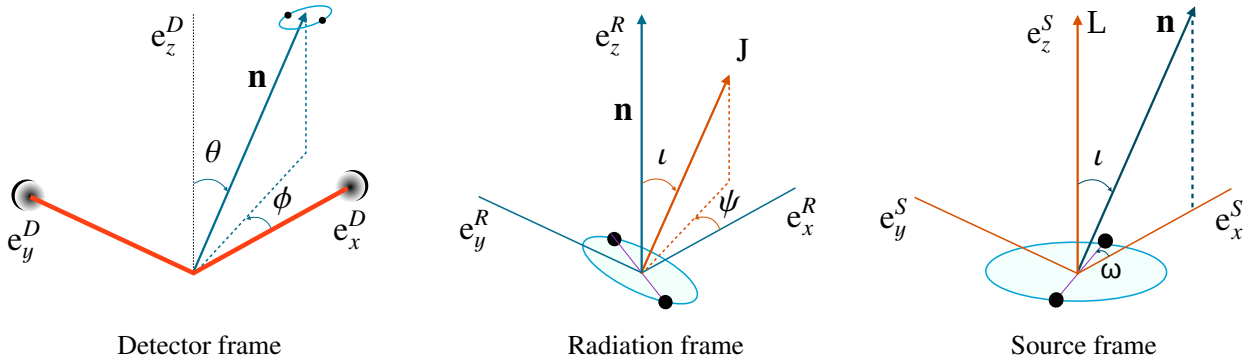


Figure 5: Illustration of the different angles used in the antenna pattern functions $F_{+,\times}$. The left figure is in the detector frame, where the x and y axis are the detector arms. The middle figure is in the radiation frame, where the x axis is perpendicular to the y axis of the detector frame, and $\mathbf{n}$ is the vector from the detector to the source. The vector (J) is angular momentum of the inspiral. The right figure is in the source frame. The relevant angles defined here are the sky angles θ, ϕ, the inclination angle ι, the polarization angle ψ and the reference phase ϕ_0 Figure from [66].

The measured strain is then given by

$$h(t) = \frac{\delta L_x - \delta L_y}{L}. \tag{41}$$

In this configuration, we only measure $h_+(t)$. If we rotate the detector arms by 45° we would measure $h_\times$. In a more realistic scenario, the detector orientation and the source location is arbitrary. Then we measure a superposition of h_+ and $h_\times$ scaled with so-called antenna pattern functions $F_{+,\times}$:

$$h(t) = F_+ h_+(t) + F_\times h_\times(t). \tag{42}$$

To determine $F_{+,\times}$ we need to consider the relative orientations of the detector and GW. The angles we need to use are defined in Fig.(5).

If we use the angles as defined in the figure, we can use projection operators to arrive at the antenna pattern functions:

$$\begin{aligned} F_+(\theta,\phi,\psi) =& \frac{1}{2}(1+\cos(\theta)^2)\cos(2\phi)\cos(2\psi) - \cos(\theta)\sin(2\phi)\sin(2\psi), \\ F_\times(\theta,\phi,\psi) =& \frac{1}{2}(1+\cos(\theta)^2)\cos(2\phi)\sin(2\psi) + \cos(\theta)\sin(2\phi)\cos(2\psi). \end{aligned} \tag{43}$$

If we combine Eq.(42) and Eq.(43) we arrive at the generic expression for GW strain, including all the relevant parameters. The expression we end up with is

$$h(t) = F_+(\theta,\phi,\psi)h_+(\boldsymbol{\theta}_{in},t) + F_\times(\theta,\phi,\psi)h_\times(\boldsymbol{\theta}_{in},t), \tag{44}$$

with $\boldsymbol{\theta}$ the additional parameters of the inspiral. These additional parameters include the component masses, spin vectors, tidal deformabilities, phases and luminosity distance of the source. Usually, one makes the distinction between the intrinsic and extrinsic parameters

of the source. The intrinsic parameters include the source properties, which are the masses, the tidal deformabilities, and the spins. The extrinsic parameters include the additional parameters which are not intrinsic to the source, such as the sky angles, luminosity distance, inclination angle and the phases.

2.3.4 Detecting gravitational waves

An important quantity in claiming detections in GW science is the signal-to-noise ratio (SNR) [67]. This quantity is used to quantify how 'loud' a given signal is compared to the noise in the detector. First, we define an inner product between two arbitrary functions $\hat{a}(f)$ and $\hat{b}(f)$ that will be useful in defining the SNR:

$$\begin{aligned}\langle a \mid b\rangle &= \int_{-\infty}^{\infty} \frac{\hat{a}^*(f)\hat{b}(f) + \hat{a}(f)\hat{b}^*(f)}{S_n(f)} df, \\ &= 4\mathrm{Re}\int_0^{\infty} \frac{\hat{a}^*(f)\hat{b}(f) df}{S_n(f)},\end{aligned} \tag{45}$$

with $S_n(f)$ the power spectral density of the noise. This quantity is dependent on the detector, and tells us how sensitive it is to a certain frequency. We can also consider only a specific range in frequency space by replacing $0 \to f_{\min}$ and $\infty \to f_{\max}$. $f_{\min}$ and $f_{\max}$ refer to the minimum and maximum frequencies seen by the detector. Now we define the SNR [67] as

$$\rho = \frac{\langle d \mid h\rangle}{\sqrt{\langle h \mid h\rangle}} \tag{46}$$

with d the data from the detector and h a template waveform with certain parameters. The tildes represents a Fourier transform and $*$ represents a complex conjugation. $S_n(f)$ is the noise power spectral density (PSD), which describes the power of the noise of the detectors at different frequencies. The minimum frequency is the lowest sensitive frequency of the detector and the maximum frequency is the maximum frequency reached by the template. If the waveform is the same as the signal found in the data, i.e. $d = h + n$ where n is the noise, the SNR is expected to be high. If we use a template that is different to the signal in the data or if there is no signal at all, the SNR is low.

If we consider a network of detectors, the total SNR of the network is defined by the quadratic sum of the detector SNRs:

$$\rho_{\mathrm{net}} = \sqrt{\sum_{i=1}^{N_{\mathrm{det}}} \rho_i^2} \tag{47}$$

with ρ_{net} the network SNR, ρ_i the SNR of a single detector and N_{det} the number of detectors in the network. In practice, the SNR is in part used to claim detections by utilizing a so-called template bank [68]. This consists of multiple template signals, which are then matched with the detector output to calculate the SNR of each template. An example template bank is shown in Fig.(6). If the SNR of a template matching with the data is higher than a certain value (usually 4), we calculate other detection statistics, which are different for each detection pipeline. This detection algorithm is also called matched filtering.

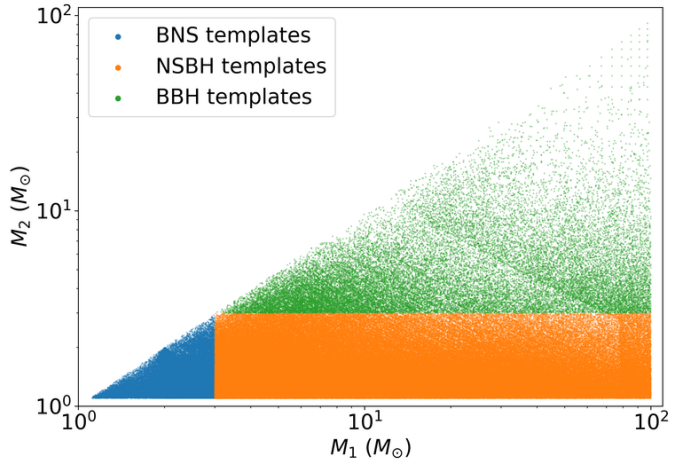


Figure 6: Representation of a template bank as used by [69]. Each point represent a signal template, classified by their component masses. The colours represent the sub-banks the template bank is divided into, corresponding to the numbers in the plot. The BNS range has a more dense population of templates because BNS signals are longer, which means that small phase differences between signals have more time to accumulate, increasing the mismatch. A template is mismatched if a template matches with a signal with significantly different source parameters.

It is also useful to define the optimal SNR to characterize the loudness of a signal without injecting it into the noise. The optimal SNR is found by matching the template with itself and is given by

$$\rho_{\text{opt}} = \sqrt{\langle h \mid h \rangle} \tag{48}$$

In essence, this boils down to neglecting the noise effects on the SNR, which makes the calculation of it easier.

2.3.5 Partial inspiral detection

In this work, we would like to characterize pre-merger BNS signals. Therefore, the full SNR of the signal is not useful to characterize the loudness, but we instead use the partial inspiral SNR (PISNR) as defined by G. Baltus et al. in [70, 71]. Instead of integrating over the full frequency range like in Eq.(48), the PISNR uses the same equation but integrated until a specific maximum frequency called the cut frequency, f_{cut}.

The frequency evolution is described up to first order by Eq.(18), shown in Fig.(7). The cut frequency corresponds to the maximum frequency reached by the signal at a certain time before the merger. Because the components of the signal are not evenly distributed along the frequency space, the PISNR is not linearly dependent on the frequency. Fig.(8) shows the normalized PISNR dependent on the frequency[10]. As we will discuss in more detail in section 4, we train neural networks on constant cut frequencies.

[10]The figure focuses on the frequency range considered in this work.

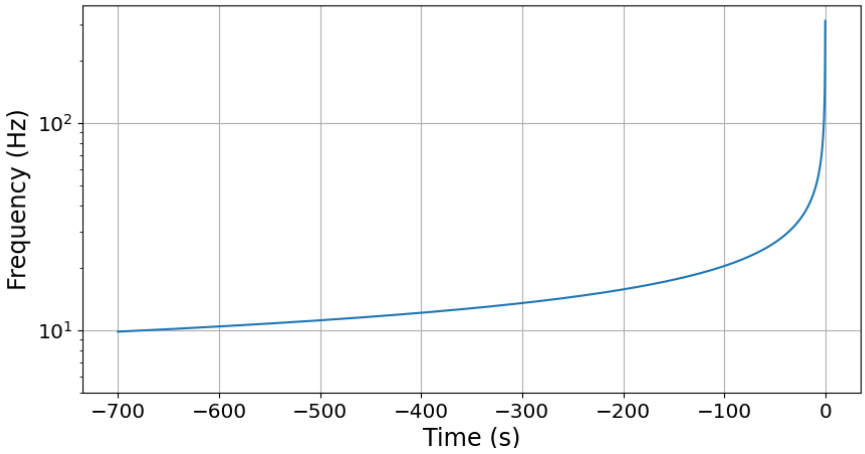


Figure 7: Frequency evolution of the first order approximation of a waveform with $\mathcal{M}_c = 1.8M_\odot$ as described in Eq.(18). $t = 0$ corresponds to the time where the approximation breaks down.

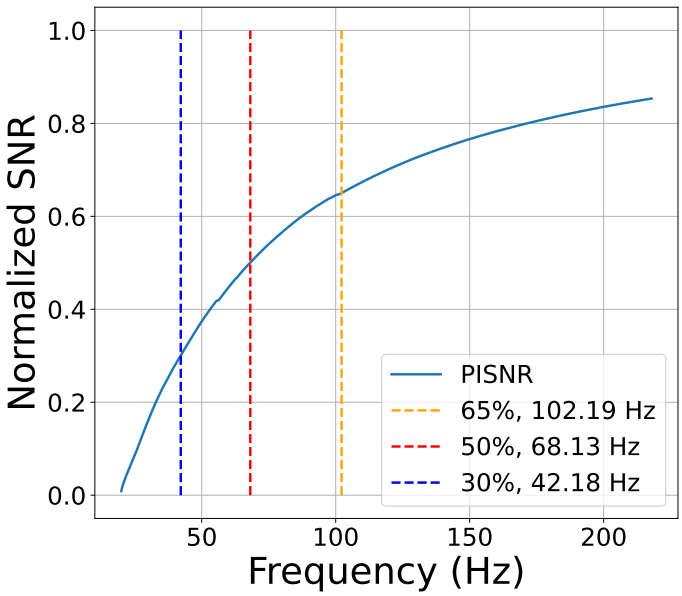


Figure 8: Normalized PISNR as a function of the frequency in the frequency range considered in this work. The vertical lines show where the PISNR reaches 30%, 50% and 65% of the total SNR, which happens at about 40, 70 and 100 Hz respectively.

2.3.6 Parameter estimation

If a detection happens, we want to estimate the parameters of the signal as well as possible. To do this, we use Bayes' theorem. Bayes' theorem is given by

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)}. \tag{49}$$

where $p(A \mid B)$ is defined as the probability of event A given event B. This theorem can be used to define conditional probabilities that can be used to estimate the properties of a signal in noise. In Bayesian parameter estimation, we would like to calculate the probability of observing a signal with certain parameters $\boldsymbol{\theta}$ given the data d and assuming the signal is from a certain type of source, such as a BNS (which we call the hypothesis $\mathcal{H}$). This is equivalent to calculating $p(\boldsymbol{\theta} \mid d, \mathcal{H})$. This probability distribution is called the *posterior probability distribution* or the posterior for short. By using Eq.(49) we find the expression

$$p(\boldsymbol{\theta} \mid d, \mathcal{H}) = \frac{p(d \mid \boldsymbol{\theta}, \mathcal{H})p(\boldsymbol{\theta} \mid \mathcal{H})}{p(d \mid \mathcal{H})}. \tag{50}$$

$p(d \mid \boldsymbol{\theta}, \mathcal{H})$ is called the *likelihood*, which is still unknown. $p(\boldsymbol{\theta} \mid \mathcal{H})$ is the *prior probability distribution*, which can be described as the probability that a certain parameter set $\boldsymbol{\theta}$ is found under a certain hypothesis. Generally, the priors are taken to be as agnostic as possible to avoid biasing the results on assumptions, but they do depend on the hypothesis. For example, if we expect the signal source to be a BNS, we do not expect the chirp mass to be higher than $2.21M_{\odot}$, since this is the maximum chirp mass limit for BNS signals. $p(d \mid \mathcal{H})$ is the *evidence* for the hypothesis $\mathcal{H}$. It plays the role of a normalization factor. From now on we will drop $\mathcal{H}$ from the expression for convenience. Next, we want find an expression for the likelihood. We do this by investigating the composition of the data in question.

The data we want to investigate is of the form $d = h(\boldsymbol{\theta}) + n$ with $h(\boldsymbol{\theta})$ the signal with parameters $\boldsymbol{\theta}$ and n the noise.

The detector outputs a digitized strain. The output of the detector in frequency domain in absence of a signal is

$$(\hat{n}_0, \hat{n}_1, \hat{n}_2, \cdots, \hat{n}_N) \tag{51}$$

with $\hat{n}_i = \hat{n}(f_i)$ the discrete Fourier transform of the noise. If we take the continuum limit and assume the noise is (colored) Gaussian noise, we can use the definition of the inner product in Eq.(45) to find the probability of observing a certain noise realization as

$$p[n] = \mathcal{N} e^{-\frac{1}{2}\langle n|n\rangle} \tag{52}$$

with $\mathcal{N}$ the normalization factor and $S_n(f)$ the noise power spectral density (PSD). If we now use the expression $n = d - h(\boldsymbol{\theta})$ and insert it into Eq.(52) we get the expression for the likelihood:

$$p(d \mid \boldsymbol{\theta}, \mathcal{H}) = \mathcal{N} e^{-\frac{1}{2}\langle d - h(\boldsymbol{\theta};f) | d - h(\boldsymbol{\theta};f)\rangle}. \tag{53}$$

This equation allows us to evaluate the likelihood of a single point in the parameter space. In order to map out the likelihood fully, one needs to calculate this integral to map out the multidimensional parameter space. Because of the high dimensionality of the parameter space, this is prohibitively expensive to calculate in every point of the parameter space.

Instead, techniques like *nested sampling* [28] and *Markov chain Monte Carlo* [27] are used to increase the efficiency of exploring the parameter space while retaining accuracy. These methods are then used by estimation software such as `LALInference` [72], `PyCBC inference` [73] and `Bilby` [74] to estimate the parameters of a GW signal. `LALInference` is a pioneering software package that implements nested sampling and MCMC in the context of GW parameter estimation, while `Bilby` is a more modern user-friendly implementation. However, even with optimizations it can still take weeks to explore the likelihood for a single signal fully. Because of this, we instead employ a ML algorithm to approximate the posterior directly, which we will introduce in section 3.

2.4 Multi messenger astronomy

Now that we have discussed the general concepts of GW generation and detection, we focus more on the specific science case we want to investigate, namely , multi messenger astronomy (MMA) using GW sources. This section is organized as follows: first we give a short description of neutron stars in GW science, then we discuss the GW170817 detection and its contributions to the scientific community. Next we discuss early detection and fast sky localization of BNS signals and then we motivate using ML for sky localization.

2.4.1 Neutron stars

Neutron stars are stellar objects which remain after a super giant star collapses at the end of its life cycle [75]. They are very compact: neutron stars have a radius of $\mathcal{O} \sim 10$ km and a mass on the order of $\mathcal{O} \sim 1.5 M_\odot$ [76]. Most models imply that they are composed almost entirely of neutrons, because the extreme pressure of the formation caused the electrons and protons to fuse together into neutrons. These stars are protected from further collapse by neutron degeneracy pressure and strong force interactions [77].

The equation of state (EOS) of neutron stars is still a much discussed field of research [78]. Because of the extremity of the matter in question, one needs to include both nuclear interactions and general relativity to be able to describe the forces that hold them together and keep them from collapsing. There are a number of different EOS theories that lead to different observable quantities, like the speed of sound, mass, radius and Love number (which relates to the tidal properties [79]) of neutron stars. Experimental observations, from for example the Neutron star Interior Composition ExploreR (NICER) telescope [80] and the GW observatories, can provide constraints to the EOS theories.

Obtaining more information about the EOS of neutron stars gives us a unique insight into their formation and composition, allowing us to probe one of the most extreme forms of matter in the universe. Because of the extreme mass of neutron stars, they can serve as a natural laboratory to test theories on quantum gravity [81].

2.4.2 Example: GW170817

On the 17$^{\text{th}}$ of August 2017, the LIGO and Virgo detectors observed a signal originating from a BNS inspiral. 1.7 seconds later, the Fermi Gamma-Ray-Burst Monitor [12] and the INTEGRAL satellite [13] observed a gamma ray burst (GRB) coming from the same

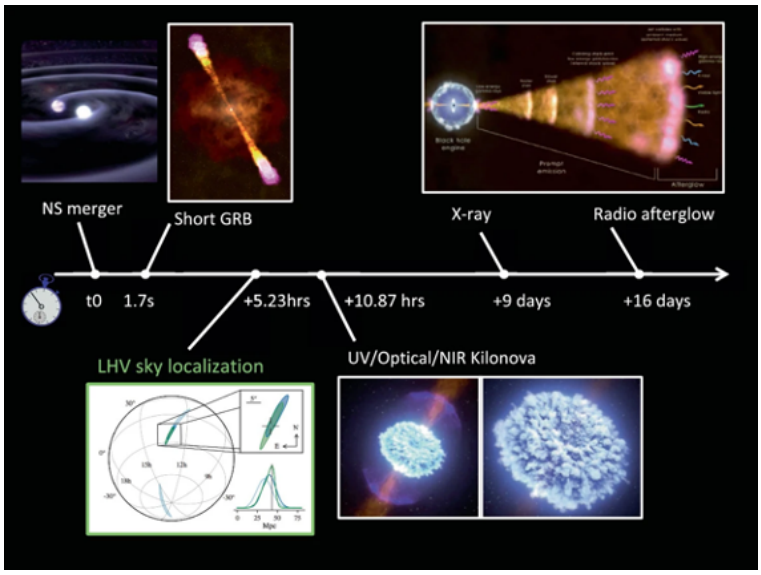


Figure 9: Follow-up observation timeline of the GW170817 detection. Figure taken from [82].

location [14], which was found out in an archival search. About 5 hours later, the LIGO and Virgo collaboration published the estimated sky location of the BNS signal which allowed for multiple electromagnetic follow-up studies. Fig.(9) shows a timeline of the detection and the follow-up studies. This detection opened up the field of MMA, which can combine observations of high-energy neutrinos, ultra-high energy cosmic rays, gamma ray bursts, other EM channels and GW observations [15]. We will now discuss some of the scientific progress made following the GW170817 detection.

One significant contribution of the GW170817 detection was a independent measurement of the Hubble constant [16]. This quantity is of fundamental importance to astrophysics since it sets the local expansion rate of the universe. So far, there have been a number of distinct measurement techniques for finding the Hubble constant. One technique calculates it using the cosmic microwave background [83]. Another technique involves using type Ia supernovae [84]. Both these methods are reasonably accurate, but they produce significantly different measurements of the Hubble constant.

The measurement done with GW170817 combines the distance to the source inferred for the gravitational wave measurement with the recession velocity inferred from measurements of the redshift using EM data to find the Hubble constant. The measurement of the Hubble constant from this source is still a bit inaccurate, but it shows reasonable promise since it will improve with more simultaneous observations of GW and GRB from neutron stars [16].

The GW170817 detection also contributed to constraining the EOS for neutron stars because it allowed for a independent measurement of the radii of the neutron stars. The measured radii of the neutron stars are $R_1 = 10.8^{+2.0}_{-1.7}$km for the heavier star and $R_2 = 10.7^{+2.1}_{-1.5}$km for the lighter star from the LIGO and Virgo data alone [17]. This, combined with the measured masses of the neutron stars provides additional constraints to EOS models because they need to predict the existence of neutron stars with these masses and radii. Because of this, some neutron star EOS theories were already excluded.

Another application of the GW170817 detection is a independent measurement of the

speed of gravity compared to the speed of light. Since we can estimate at which time the BNS merger should emit a gamma ray, depending on the EOS model, the time-of-arrival difference between the GW and the gamma ray burst provides us with a measurement of the speed of gravity [14].

The GW170817 detection has provided us with a lot of new insight, but we can do better in the future. The delays between the detection and the sky localization means that we did not observe the merger directly in the EM spectrum. If we can observe the BNS inspiral before the merger, we can do a lot of additional science with the observation.

2.4.3 Early detection

To discover even more interesting physics from BNS detections, we want to create techniques that can detect and localize the BNS inspiral *before* the merger. This will allow us to study additional physics, which we will discuss now.

In addition to possessing some of the highest densities of material, neutron stars also have some of the strongest magnetic fields in the universe. During the merger, these fields can interact non-trivially and establish a nearly force-free magnetosphere [85] filled with pair-plasma at the time of merger, which can lead to the emission of pulsar-like radio emissions [86], in addition to other interactions. Numerical simulations have already been done on this subject [19], which can be validated with real observations.

BNS mergers are also thought to be a astrophysical source for rapid neutron-capture (r-process) nucleosynthesis [87, 88], which is a process responsible for creating nuclei heavier than iron [89]. During BNS mergers, these elements can be created and emit spectral lines that can be observed. The GW170817 observation provided us with measurements of this. Since these emissions fade over time, early detection can provide us with more data [20].

In some cases, a BNS merger can produce a stable massive NS remnant (called a magnetar) instead of a black hole [21, 90–92]. These objects would then emit X-ray and optical signals that could be detected at the merger, which could then help us determine the state of the remnant object [93].

In conclusion, if we can do early detection, we can do even more new physics one could not do without early detection.

2.4.4 Pre-merger sky localization

The first part of doing pre-merger sky localization is detecting the inspiral before it happens via an early alert[11]. Multiple studies have shown the ability to provide early alert for BNS inspirals at design sensitivity using the LIGO and Virgo detector network. Some of these studies rely on a matched-filtering based approach [94–96] , while others use convolutional neural networks to produce triggers [70, 71, 97]. These studies can provide early alerts up to several minutes before the merger. In our work, we consider the scenario where a detection has been made by one of these early alert frameworks. We then want to rapidly estimate the sky location to give the follow-up EM telescopes a location to start the search for a EM counterpart.

[11]With early alerts we refer to a trigger that is issued *while* a BNS signal is in the sensitivity band.

There are already a number of studies that create a framework to do rapid sky localization. We will discuss the most relevant 'competitors' and talk about the challenges and differences with our framework.

The first framework we want to discuss is called BAYESTAR [29]. This framework is a matched-filtering based approach that uses the output of the template bank, namely relative SNRs, relative arrival times, and relative phases at arrival to do marginalized Bayesian parameter estimation to rapidly estimate the sky location of BNS signals. The framework is nominally used for complete signals but it can also be adapted to produce skymaps when only part of the signal is observed [29].

The second relevant framework is called GWSkyLocator [30]. This framework also uses the SNR time series of the best-matched template from the template bank and the intrinsic parameters. This information is then given to a neural network which estimates the sky location. This gives comparable results to BAYESTAR, but because it also uses the template bank as input it has some of the same challenges.

Both frameworks rely on a matched filtering based approach. For these methods to work optimally, the best-matched signal needs to be very close to the real signal. However, because the template bank consist of discrete points in the mapped parameter space, the best-matched signal is never exactly the same as the real signal. Also, the best-matched templates can be off due to non-Gaussianities in the noise, but this is a problem that arises with multiple frameworks. Matched filtering only gives a point-estimate of the intrinsic parameters which does not account for the inherent uncertainty in the estimation which can be very large in the early phase of the inspiral.

To address these challenges, we construct a ML based framework that can estimate the sky location from the data directly without requiring the input of a template bank, in addition to estimating other intrinsic parameter useful for the EM follow-up, such as the component masses, inclination angle and luminosity distance.

3 Machine learning

In this work, we use a machine-learning (ML) to do neural posterior estimation (NPE). NPE is a subclass of ML methods that is used to do parameter estimation by learning patterns in a dataset that can be used to estimate the parameters of a signal; for example using a GW data strain to estimate the chirp mass, mass ratio etc. ML methods are implemented by creating a neural network (NN), which we explain in more detail later.

In this section we give a basic introduction to the theory behind the implementation of ML networks. Next we introduce normalizing flows as a method to model probability distributions. Then we discuss methods for dimensionality reduction to preprocess the data given to the flow network. We end the chapter with a general overview of NPE, how it is used in GW science.

3.1 Introduction to machine learning

Before we discuss our implementation, we explain the basics of the construction and operation of ML networks. ML networks are implemented in structure called a neural network (NN). To produce the desired output, a NN needs to be trained to minimize a *loss function* with a *backward pass*. We explain these concepts in more detail in the coming sections.

3.1.1 Neural networks

A NN is an artificial model inspired by the structure of a brain. They consist of connected nodes, called neurons, which transfer information to other neurons to generate the desired output. This is inspired by the function of synapses in a biological brain. Neurons are grouped together in layers, with neurons from the first layer typically passing inputs to neurons of the second layer and so forth. In a NN, the signals transferred between neurons are (typically) real numbers, which are processed by the neurons and send to the next layer. Neural layers are grouped into three distinct types:

- **Input layers** are the interface between the input data and the neural network; they process the input data and pass it to the hidden layers.
- **Hidden layers** are the layers in between the input and output layers. They are called hidden layers because their interactions are usually not visible. They usually make up the bulk of the network structure.
- **Output layers** give the output of the network, depending on the requirements. For example: if one builds a neural network to classify a cat or a dog, the output layer can consist of two neurons. These neurons then output the estimated probability of the input being either a cat or a dog.

Fig.(10) shows a illustration of these layers and the connections between them. Each of the connections between neurons have an associated *weight* and *bias* and are processed with an *activation function*, which we discuss shortly.

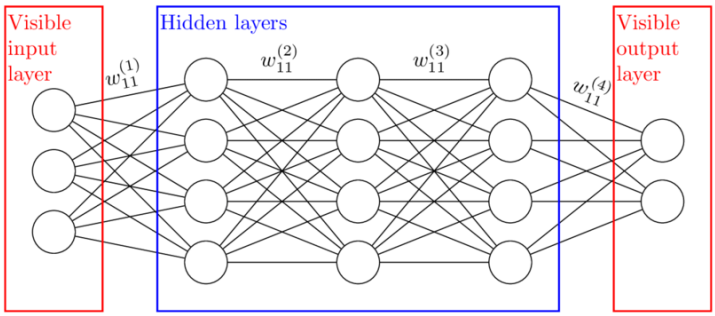


Figure 10: Structure of a simple neural network, divided in input, hidden and output layers. The circles represent the neurons, and the lines between them represent the connections between the neurons. The connections have *weights* associated with them represented by by w_{jk}^{i} Figure taken from [98].

3.1.2 Forward pass

To describe the flow of data through a neural network, we need to define how the neurons interact with each other. The transfer of information between the neurons from input to output is called a *forward pass*. The goal of our neural network is to produce a result $\mathbf{t}$ from input values $\mathbf{x}$. The output of a specific neuron, $y(\mathbf{x})$ is depended on the input from the previous neuron layer and the weights and biases associated with the connections between the previous neural layer and the neuron in question:

$$y(\mathbf{x}) = \sum_{i}^{D} w_i x_i + b_i, \tag{54}$$

where D is the dimension of the previous neuron layer, w_i are the weights and b_i are the biases associated with neuron connection. To increase the expressive power of the network it is crucial include non-linearity in the transformations, otherwise the network can only model linear transformations. This is done by including a activation function f. The output of the activation function is then passed to the neuron. This can be expressed as

$$y(\mathbf{x}) = f\left(\sum_{i}^{D} w_i x_i + b_i\right). \tag{55}$$

Including an activation function in the regression allows the network to achieve a non-linear mapping from the input to the output layers [99]. There are multiple choices one can make for what activation function to use, in our work we use the GELU [100] and the RELU [101] activation functions, given by

$$\begin{aligned} f_{\text{GELU}}(x) =& x\frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right) \\ f_{\text{RELU}}(x) =& \begin{cases} x \text{ if } x > 0, \\ 0 \text{ otherwise}, \end{cases} \end{aligned} \tag{56}$$

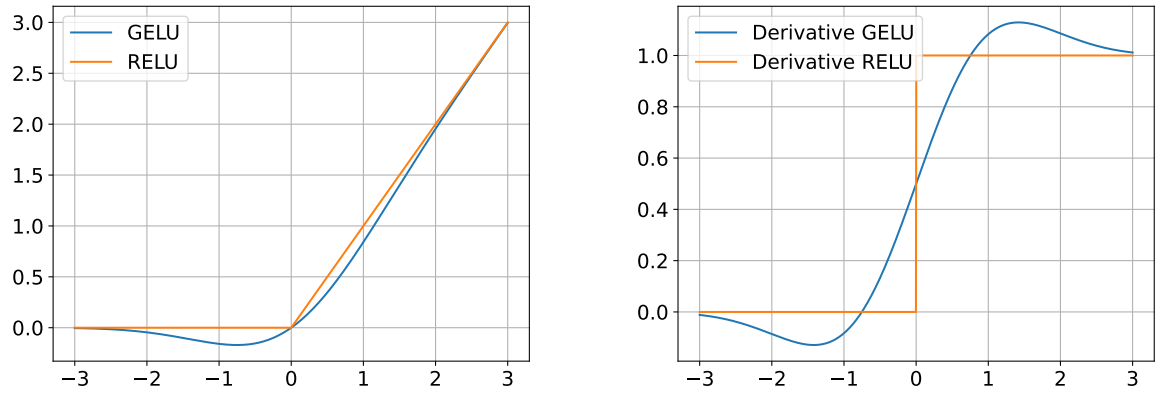


Figure 11: Comparison of the GELU and RELU activation functions and their derivatives.

with erf the Gaussian error function. The functions and their derivatives are compared in Fig.(11). As shown, the GELU function and RELU function are similar, but GELU is also continuous in its derivative, which RELU is not.

The output of the neural network can be calculated simply by repeating the operation of Eq.(55) for each neuron layer. To initialize the NN, one usually draws random values for the weights and biases. At this point the output will be nonsense; to make it more useful we need to train the network.

3.1.3 The loss function

Before we can actually train our network, we need to specify what we want it to learn. This is done by defining a *loss function*. The idea behind a loss function is to attribute a numerical value to the performance of the network, which it should try to minimize. The choice of loss function is dependent on the application you want the network to fulfill.

The goal of the NN used in this work is to do maximum likelihood estimation; i.e. estimate the parameter set with the highest likelihood as given by Eq.(50). This corresponds to finding the most probable posterior. We want to find

$$\hat{\boldsymbol{\theta}} \equiv \text{argmax}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \log p(\boldsymbol{\theta} \mid d), \tag{57}$$

which can be interpreted as the value of $\boldsymbol{\theta}$ for which the observed $\mathbf{d}$ is the most probable. $\boldsymbol{\Theta}$ refers to the allowed set of parameters for $\boldsymbol{\theta}$. If we can assume that all the training samples are independent and identically distributed over the parameter space, we can define the negative log-likelihood as

$$\mathcal{L}(\boldsymbol{\phi}) \equiv -\sum_{i}^{N} \log p(\boldsymbol{\theta}_i \mid \mathbf{d}_i, \boldsymbol{\phi}) \tag{58}$$

with N a summation over the samples i, and $\boldsymbol{\phi}$ the collection of network parameters that we wish to optimize. We use this function as the loss function of our network. To find the parameters with the highest likelihood, we then have to minimize the loss function. The loss function is minimized with a *backward pass* through the network, which we discuss next.

3.1.4 Backward pass

To optimize the output of the network, we want to change the weights and biases to a certain value that minimizes the loss function. The parameter space of the loss function depends on all the weights and biases in the network and is therefore quite large. In this section we consider a unspecified loss function.

To initialize the network, one usually assigns a random variable drawn from a normal distribution to each weight and bias. Then we start the first *training loop*; we perform a forward pass with a batch of samples from the data and calculate the loss of the network according to the results. We then want to update the weights and biases to minimize the loss. The loss function can be visualised as a hyper-dimensional plane[12] with multiple minima and maxima. The goal is to update the network such that the overall loss of the network decreases and we find one of the minima of the loss function. This is illustrated in Fig.(12) with a 2 dimensional example. In reality, the dimensionality of the loss function is way bigger. We update the weights and biases by doing a *backward pass*. During a backward pass, we update the network parameters by employing a optimizer. A widely used optimizer is stochastic gradient descent [102], given by

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \frac{\alpha}{n} \sum_{j=1}^{n} \nabla \mathcal{L}_j(\mathbf{w}_i), \tag{59}$$

where w_i represents the weights at training step i, α is the learning rate and n the amount of samples in a batch. The derivative of the loss of each sample is with respect to the weights. The same function also applies for the other network parameters, such as the biases. The learning rate and the batch size are both hyperparameters that we must choose ourselves. The learning rate parameterizes how 'fast' the network moves through the hyperplane. If we choose a learning rate that is too large, we can 'overshoot' the minima of the loss function and hinder convergence. If we choose as learning rate that is too small, we can more easily get stuck in local minima.

Optimizers are used to increase the efficiency of the gradient descent process, by helping us avoid local minima in the hyperplane of the loss function. An optimizer works by modifying Eq.(59) to increase the training efficiency. There are a number of optimizers we can choose from, but in this work we work with the Adam optimizer [104].

The Adam optimizer is based on the concept of estimating the 'momentum' of the gradient descent algorithm, which we can explain with an analogue. If we consider a ball rolling down a bumpy hill, the future location of the ball depends on the current location, current momentum and current acceleration. Because the ball has a nonzero momentum and acceleration it will not get stuck on small potholes in the way, but continue rolling down the hill. If we consider the normal gradient descent algorithm, the ball would roll down the hill a bit, stop, and roll again. This makes it way more likely to get the ball stuck in a hole. This concept of momentum is implemented by the Adam optimizer by estimating and implementing the momentum of the descent.

[12]Each network parameter is one dimension of the network. The loss function is a function of all the network parameters, so it is a very high-dimensional function

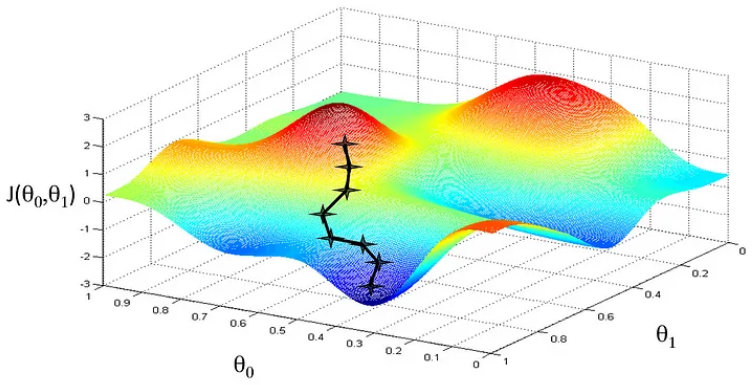


Figure 12: Representation of the process of gradient descent. The loss function $J(\theta_0, \theta_1)$ depends on two parameters: θ_0 and θ_1. The x, y values of the black dot at the top of the hill represent the network parameters, and the height represents the value of the loss function at the start of training. After each training loop, we update the parameters iteratively, ending up in a minimum of the loss function. The figure shows this as the dots that follow the first one: each dot represents a different value of the network parameters, which descend along the hill of the loss function. Figure taken from [103].

3.2 Normalizing flows

To create NN that can estimate probability distributions, we use a ML framework called normalizing flows (NFs) [105]. A NF operates by modelling invertible transformation functions that iteratively transform samples from a simple distribution to the desired complex distribution. In the following sections we discuss the constituent parts of a NF network and how they operate.

Since we get a bit deeper into the mathematical expressions we want to clarify the notation: in the coming sections an capital letter X refers to a matrix, and a bold letter $\mathbf{x}$ refers to a vector of some kind.

3.2.1 Transformations

To start our discussion on NFs, we need to talk about the type of transformations the networks are based on. For a more in-depth discussion, we refer the reader to [106]. We will start with a mathematical description of the transformation we want to implement. Let $\mathbf{x}$ be a D-dimensional real vector with elements from the allowed parameter space. We want to create a joint distribution over $\mathbf{x}$, called $p_x(\mathbf{x})$. The main idea of a flow-based model is sample from this joint distribution by using a transformation T of a real vector $\mathbf{u}$ sampled from a distribution $p_u(\mathbf{u})$

$$\mathbf{x} = T(\mathbf{u}) \quad \text{with} \quad \mathbf{u} \sim p_u(\mathbf{u}). \tag{60}$$

We define $p_u(\mathbf{u})$ as the *base distribution*. T depends on the network parameters $\boldsymbol{\phi}$.

In NFs, we require the transformation T to be *invertible* and both T and T^{-1} need to be *differentiable*. If these conditions are met, the probability distribution $p_x(\mathbf{x})$ is well defined and can be found by a change of variables

$$p_x(\mathbf{x}) = p_u(\mathbf{u})|\det J_T(\mathbf{u})|^{-1} \quad \text{where} \quad \mathbf{u} = T^{-1}(\mathbf{x}) \tag{61}$$

We can also write $p_x(\mathbf{x})$ in terms of the Jacobian of T^{-1}

$$p_x(\mathbf{x}) = p_u\left(T^{-1}(\mathbf{x})\right)\left|\det J_{T^{-1}}(\mathbf{x})\right| \tag{62}$$

The Jacobian matrix $J_T(\mathbf{u})$ is defined as

$$J_T(\mathbf{u}) = \begin{pmatrix} \frac{\partial T_1}{\partial u_1} & \cdots & \frac{\partial T_1}{\partial u_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_D}{\partial u_1} & \cdots & \frac{\partial T_D}{\partial u_D} \end{pmatrix} \tag{63}$$

The idea behind a NF model is to parameterize the transformations T (or T^{-1}) with a neural network and choosing $p_u(\mathbf{u})$ to be simple distribution, usually a multivariate Gaussian distribution.

The transformation function T can be interpreted as a warping of the space $\mathbb{R}^D$ which is conditioned such that samples of $p_u(\mathbf{u})$ transform into samples of $p_x(\mathbf{x})$. The absolute Jacobian determinant $|\det J_T(\mathbf{u})|$ quantifies the relative change in volume of a small neighbourhood around $\mathbf{u}$ due to T. Because the probability mass of the samples is conserved, the transformation can only change the probability density in the $\mathbb{R}^D$. The inverse transformation T^{-1} instead transforms samples of $p_x(\mathbf{x})$ into samples of $p_u(\mathbf{u})$[13].

Another important property of invertible and differentiable transformations is that they are *composable*, which means that a chain of transformations, $T_2 \circ T_1$, is also invertible and differentiable. The inverse and the determinant of the Jacobian are given by

$$\begin{aligned} (T_2 \circ T_1)^{-1} &= T_1^{-1} \circ T_2^{-1} \\ \det J_{(T_2 \circ T_1)}(\mathbf{u}) &= \det J_{T_2}(T_1(\mathbf{u})) \cdot \det J_{T_1}(\mathbf{u}) \end{aligned} \tag{64}$$

These properties allow us to link simple transformations together to increase the expressive power of the NFs. This results in a transformation chain $T = T_K \circ \cdots \circ T_1$ where each T_k transforms $\mathbf{z}_{k-1}$ into $\mathbf{z}_k$ assuming $\mathbf{z}_0 = \mathbf{u}$ and $\mathbf{z}_K = \mathbf{x}$. As discussed before, the inverse flow $T_1^{-1} \circ \cdots \circ T_K^{-1}$ transforms samples from $p_x(\mathbf{x})$ into samples from $p_u(\mathbf{u})$.

In practice, the flow model has two modes: 1) sampling from the distribution modelled by the NF via Eq.(61) and 2) calculating the likelihood of samples using Eq.(62), which is used in training the network. These modes have different computational requirements. Sampling requires us to sample from $p_u(\mathbf{u})$ and compute the forward transformation by using T. Evaluating the model requires us to compute the inverse transformation T^{-1} and its Jacobian, and evaluate the density of $p_u(\mathbf{u})$. As we will discuss later, these modes are the sampling and training modes. The construction of the transformations therefore affects the efficiency of the training and sampling of the network. In our work, we would ideally like *both* to be reasonably fast; we the training to be fast, but we also want efficient sampling. In the next section we discuss how to implement the transformations of a NF in practice.

3.2.2 Flow structure

To implement T_k or T_k^{-1} we use a model with parameters ϕ_k, which we will denote as f_{ϕ_k}. The choice of which transformation to implement is dependent on the intended usage. In

[13] As we will discuss later, the inverse transformation is used in training the NF.

our work we implement the inverse transformation to make training more efficient. In any case, we need to make sure that the calculation of the determinant of the Jacobian of T_k and T_k^{-1} is tractable[14], which restricts our choice of how to implement f_{ϕ_k}. In our work, we use a coupling flow [107] to additionally ensure that the sampling and training is fast. This is done by placing certain requirements on the structure of the determinant.

To simplify the notation a bit, from now on we drop the dependence of the model parameters on k, refer to the input of the model as $\mathbf{z}$ and the output as $\mathbf{z}'$.

In general, the flow layers are implemented by specifying f_ϕ to have the form

$$z_i' = \tau(z_i; c_i(\mathbf{z}_{<i})), \tag{65}$$

where τ is called the *transformer* and c_i the i-th *conditioner*. The transformer is a monotonic function of z_i (which makes it invertible) and parameterized by $c_i(\mathbf{z}_{<i})$. The transformer specifies how the flow acts on z_i to produce the output z_i'. We discuss our choice of transformer in section 3.2.3. The conditioner is a function with the constraint that the i-th conditioner can only depend on the variables with a dimension index of less than i.

In our work, we use a *coupling flow* conditioner. This conditioner is *computationally symmetric*, i.e. equally fast to evaluate or invert. This is implemented by choosing an index d (typically $D/2$, with D the total amount of transformed samples). The coupling layer splits $\mathbf{z}$ into two parts $\{\mathbf{z}_{\leq d}, \mathbf{z}_{>d}\}$. Then, the conditioner is designed such that

- The conditioners $c_{i\leq d}$ are identity functions, so they do not depend on $\mathbf{z}$.
- The conditioners $c_{i>d}$ are functions that only depend on $\mathbf{z}_{\leq d}$, so $c_{i>d}(\mathbf{z}_{\leq d})$.

The samples $\mathbf{z}_{\leq d}$ are transformed element-wise and do not depend on any other samples, but the transformation of the samples $\mathbf{z}_{>d}$ depends on $c_{i>d}(\mathbf{z}_{\leq d})$. Because the conditioners $c_{i<d}$ do not depend on the samples it transforms, it can be evaluated in parallel for each sample. If we additionally fix the transformers for the samples $\mathbf{z}_{\leq d}$ to the identity function, we can express the flow functions as

$$\begin{aligned} z_i' &= z_i \quad \text{for} \quad i \leq d, \\ z_i' &= \tau(z_i; c_i(\mathbf{z}_{\leq d})) \quad \text{for} \quad i > d. \end{aligned} \tag{66}$$

The inverse is then given by

$$\begin{aligned} z_i &= z_i' \quad \text{for} \quad i \leq d, \\ z_i &= \tau^{-1}(z_i'; c_i(\mathbf{z}_{\leq d})) \quad \text{for} \quad i > d. \end{aligned} \tag{67}$$

These transformations are illustrated in Fig. (13). This flow structure has a Jacobian of the form

$$J_{f_\phi} = \begin{pmatrix} I & \mathbf{0} \\ \frac{\partial \tau(\mathbf{z}_{i>d}; c_i(\mathbf{z}_{\leq d}))}{\partial \mathbf{z}_{j\leq d}} & \frac{\partial \tau(\mathbf{z}_{i>d}; c_i(\mathbf{z}_{\leq d}))}{\partial \mathbf{z}_{j>d}} \end{pmatrix} \tag{68}$$

[14]Some clarification of a 'tractable' Jacobian determinant calculation: any Jacobian with a dimension of $D\times D$ can be calculated normally in $\mathcal{O}\left(D^3\right)$ operations. For most flow based models, the Jacobian computation time should be at most $\mathcal{O}\left(D\right)$. This can be achieved by making specific choices for the implementation of f_{ϕ_k}

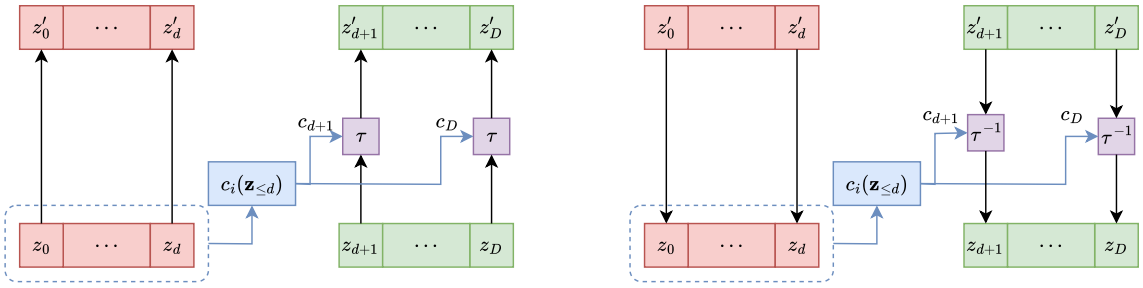


Figure 13: Illustration of the transformations done by a coupling flow layer. On the left, we show a forward transformation, and on the right we show an inverse transformation.

where I is a $(D-d)\times(D-d)$ dimensional identity matrix. The determinant of the Jacobian is then simply the product of the diagonal elements of $\frac{\partial\tau(\mathbf{z}_{i>d};c_i(\mathbf{z}_{\leq d}))}{\partial\mathbf{z}_{j>d}}$. The log-absolute determinant of the Jacobian can then be expressed as

$$\log|\det J_{f_\phi}| = \sum_{i=d}^{D}\log\left|\frac{\partial\tau}{\partial z_i}(z_i;c_i(\mathbf{z}_{\leq d}))\right|. \tag{69}$$

This makes the evaluation of the Jacobian very fast: only $\mathcal{O}(D-d)$ calculations needed, which is the same for the inverse. This efficiency does come at a cost to expressive power: we need to chain together multiple coupling layers to increase the expressivity. When constructing a flow with multiple coupling layers, the elements of $\mathbf{z}$ need to be permuted so that every sample will be transformed by the flow as well as interact with each other.

At this point, the conditioner c_i is still an arbitrary function depending on the input parameters $\mathbf{z}_{\leq d}$. This function is implemented as a neural network, which can then be trained to produce the desired outputs.

3.2.3 Bernstein polynomials

One key challenge of using NFs for probabilistic modelling is the dealing with the noise inherent in the data. Because NFs are a nonlinear model, they can be susceptible to numerical instabilities; some noise features might be amplified which causes the output of the model to be nonsensical. This can result in the amplifying of initial errors, out-of-distribution sample generation or poor generalization to unseen data. To combat this issue, we implement Bernstein-type polynomials as a transformer function, based on the work of S. Ramasinghe et al. [108]. The robustness of Bernstein-type NFs follows from the *optimal stability* of the Bernstein basis [109, 110]. We now give the definition of the Bernstein polynomial basis and list some of its features.

A n-th degree Bernstein polynomial is defined as

$$B_n(x) = \sum_{k=0}^{n}\alpha_k\binom{n}{k}x^k(1-x)^{n-k} \quad \text{with} \quad x\in[0,1]\,, \tag{70}$$

Figure 14: Illustration of the transformations done in a conditional coupling flow layer. On the left, we show a forward transformation, and on the right we show an inverse transformation.

where $\alpha_k, 0 \leq k \leq n$ are some real constants. In practice, using a higher degree of Bernstein polynomial increases the expressive power of the network.

The Bernstein polynomials are only defined with an input range of $[0, 1]$, so they are paired with a linear map from the desired interval to $[0, 1]$.

To make sure our Bernstein polynomial is invertible, we need to ensure that the Bernstein polynomial is monotonically increasing. This can be ensured by using a specific choice for the shape of the α_k parameters. In our work, we define the parameters as

$$\alpha_k = |v_1| + \cdots + |v_k|, \tag{71}$$

with v_i trainable parameters and $\alpha_0 = 0$. The Bernstein polynomial consists of a summation of functions, so the first function will have a prefactor of $|v_1|$, the second one $|v_1| + |v_2|$ etc. After each iteration, we linearly scale α_k such that $\alpha_n = 1$.

Once we ensure the inverability, we can calculate the inverse; at each iteration, given x we solve for $z \in [0, 1]$

$$B_n(z) = \sum_{k=0}^{n} \alpha_k \binom{n}{k} z^k (1-z)^{n-k} = x \Leftrightarrow \sum_{k=0}^{n} (\alpha_k - x) \binom{n}{k} z^k (1-z)^{n-k} = 0 \tag{72}$$

This equation can be solved by employing a root-finding algorithm, and with this we have everything we need to use the function as a transformer in our NF.

3.2.4 Conditional normalizing flows

So far, we have discussed a framework that can model a probability distribution by taking samples from the base distribution and transforming them into the desired probability distribution. However, in our work we want to create a *conditional* probability distribution; i.e. a distribution that depends on the input BNS strain. To model this, we create a *conditional normalizing flow*. Define the condition $\mathbf{d}$ and the target parameters $\mathbf{x}$. The NF is trained to represent the likelihood $p_{\mathbf{x}|\mathbf{d}}(\mathbf{x}|\mathbf{d})$ using a base distribution $p_u(\mathbf{u})$ and the transformation

functions f_ϕ, modelled by the NF and *conditioned* on $\mathbf{d}$. The likelihood can then be expressed as

$$p_{\mathbf{x}|\mathbf{d},\phi}(\mathbf{x}|\mathbf{d}) = p_u(f_\phi(\mathbf{u}|\mathbf{d}))\left|\frac{\partial f_\phi(\mathbf{u}|\mathbf{d})}{\partial \mathbf{x}}\right|. \tag{73}$$

In practice, the conditioning of the NF is implemented by simply giving the conditioner and transformer $\mathbf{d}$ as an additional input. This is illustrated in Fig.(14) Because the input data is strain is quite long and noisy, it is inefficient to directly input the unprocessed strain in the network. To reduce the dimensionality and reduce the noise, we use a *context network* that extracts the most important features of the data stream. Next section goes into detail about this context network and its operation.

3.3 Context network

Before we pass the generated strain into the network for conditioning, we give it to the context network for some preprocessing. The output of the context network is called the *context* used by the NF. The first part of the context network is a method called Singular Value Decomposition (SVD), which extracts important features of the data. The second part is a *residual network*, which provides some additional dimensionality reduction of the features provided by the SVD.

3.3.1 Singular Value Decomposition

In order to reduce the dimensionality and reduce noise, we want to find a more efficient representation of the BNS signals. To do this, we use a technique called Singular Value Decomposition (SVD). SVD was first introduced by V. Klema et al. [111]. The idea behind it is to provide a analogue to the eigenvalue decomposition of a $N \times N$ sized matrix and generalize it to $N \times M$ matrices.

To create a SVD decomposition that captures the important features of the BNS signals, we must first compose a template matrix H that contains an accurate representation of the signals. To do this, we generate 10000 template signals with the expected variations in the signal parameters, without any noise. The template matrix is then of the form

$$H = \{h_1, h_2, \cdots, h_M\} \tag{74}$$

Each signal h_i is a complex-valued noiseless waveform in frequency domain. We choose $M = 10,000$ to ensure that we have a good representation of the possible waveforms. Using SVD, we can factor H as

$$H_{ij} = \sum_{k=0}^{N} v_{ik}\sigma_k u_{kj}. \tag{75}$$

Here, v_{ik} is a orthogonal matrix of reconstruction coefficients, σ_k is a vector, whose elements are the singular values ranked in order of importance. u_{kj} is a matrix of orthonormal bases, whose rows are the basis vectors $\vec{u}_k$. The initial basis vectors have larger 'eigenvalues' and thus represent a large variety of the data, because the template signals have a similar structure. So, to reduce the dimensionality, we can choose to include the basis vectors that contain the

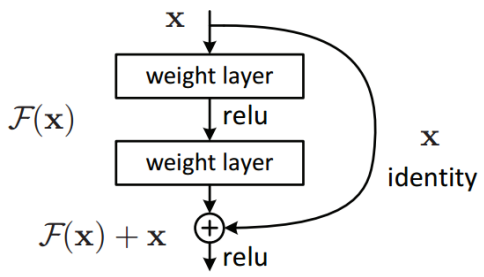


Figure 15: Residual layer. Figure taken from [112].

majority of the structure of the template signals. We use the basis vectors up to a certain limit $N' < N$. The representation of the template matrix then becomes

$$H_{ij} \approx H'_{ij} := \sum_{k=0}^{N'} v_{ik}\sigma_k u_{kj} \tag{76}$$

We refer to the collection basis vectors we use as the *SVD kernels*. The amount of SVD kernels that are necessary to represent the data accurately is dependent on the dimension of the waveforms h, so if we use a different waveform length we also need to adjust the amount of SVD kernels used.

To use the chosen SVD kernels for dimensionality reduction, we take the complex inner product between the SVD kernels and the input strain signal. The result of this inner product can be interpreted as the 'strength' of the feature represented by the SVD kernel in the signal. Since this results in a complex number, we separate the real and complex parts and pass it to the next part of the context network.

3.3.2 Residual Network

After using the SVD kernels to more efficiently represent the waveforms, we want to do some additional processing to further reduce the dimensionality of the data before giving it to the flow network. To do this, we pass the data through a residual network (ResNet). This network is composed of multiple layers of residual blocks [112]. Next, we give a short explanation on the concept behind residual layers and why we use them.

Suppose we want a neural network to approximate a certain function $\mathcal{H}(x)$, with x the input of the neural network. If $\mathcal{H}(x)$ is sufficiently complex, we need to add a increasing number of layers to the neural network to maintain the accuracy. However, at some point we run into an issue: adding more layers gives diminishing returns in increasing the accuracy [113, 114]. This is known as *degradation*. To avoid this problem, we use a ResNet.

A ResNet works as follows: instead of learning $\mathcal{H}(x)$ directly, the network learns the residual, $\mathcal{F}(x) = \mathcal{H}(x) - x$. Fig.(15) shows an illustration of a simple residual layer. The weights layers contain the trainable weights that approximate the function $\mathcal{F}(x)$. After the weight layers, we add x again so we end up with $\mathcal{H}(x)$. Using this layer construction ensures that the deeper layers of the network receives information on the original input, which increases the expressive power of the network [112].

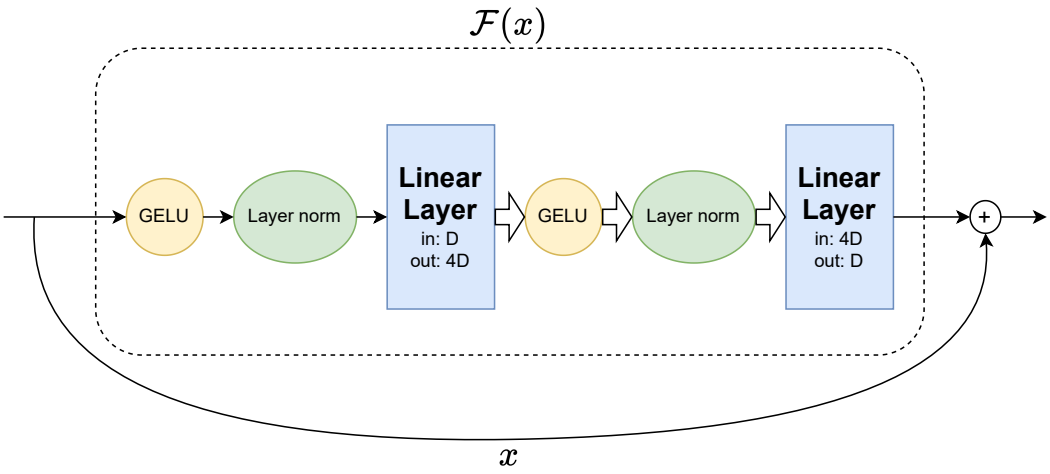


Figure 16: ResNet block scheme we use in this work.

The ResNet used in our work is implemented with an 'accordion' shape, which is constructed as follows. Each ResNet block is constructed as seen in Fig.(16). First, the array signal (of length D) is processed by a GELU activation function. Then it passes through layer normalization [115]; we calculate the mean and standard deviation of the input array, and normalize the array with the calculated mean and standard deviation, which helps with stabilizing the signal. Next the signal passes through a linear layer with an output dimension of $4D$. After that, it once again passes through a GELU activation and layer normalization. Then it passes through another linear layer, but this time with an input dimension of $4D$ and an output dimension of D. The complete ResNet is constructed out of multiple ResNet blocks. By repeatedly expanding and contracting the dimensionality, the accordion shape helps the ResNet extract important features from the data .

3.3.3 Simulation-based inference

In recent times, the field of simulation-based inference has gained increased prominence as a application of ML in science [116]. In Bayesian parameter estimation, the goal of simulation based inference is to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathbf{d}_{\rm obs})$. Classical methods such as MCMC [27] and nested sampling [28] explore the posterior distribution by directly calculating the likelihood of points in the posterior distribution, and usually try to map the parameter space with the maximum likelihood to approximate the posterior. In cases where the parameter space is high-dimensional, such as in GW parameter estimation, this method can take days to weeks to complete. In cases where we want to find the posterior as fast as possible, such as with pre-merger sky localization, these classical methods become unusable.

With simulation-based inference, we can simulate the posterior distribution from Bayes' theorem to draw samples from the posterior directly. In particular, in neural posterior esti-

mation (NPE) we use neural networks as a surrogate model to simulate the posteriors. The goal of NPE is to create a network that approximates a probability distribution

$$q_\phi(\boldsymbol{\theta}|\mathbf{d}) \approx p(\boldsymbol{\theta}|\mathbf{d}), \tag{77}$$

where $q_\phi(\boldsymbol{\theta}|\mathbf{d})$ is the probability distribution modelled by the network, dependent on the network parameters ϕ, and $p(\boldsymbol{\theta}|\mathbf{d})$ is the posterior distribution. In NPE, the neural network learns a mapping from given data $\mathbf{d}$ to approximate $q_\phi(\boldsymbol{\theta}|\mathbf{d})$. To approximate the posterior correctly, the network needs to be trained to find the best-fit network parameters $\hat{\phi}$. Given we have a sufficiently expressive network that is trained on enough data, the approximated posterior should converge to the real posterior.

In principle, any density estimation model that can return a likelihood can be used in NPE. In this work, we choose to use a conditional NF network. In GW science, ML has been used in a lot of different contexts, from glitch classification to detection [117, 118]. NPE in particular has seen a increasing amount of attention in data analysis tasks [31, 32, 34, 35].

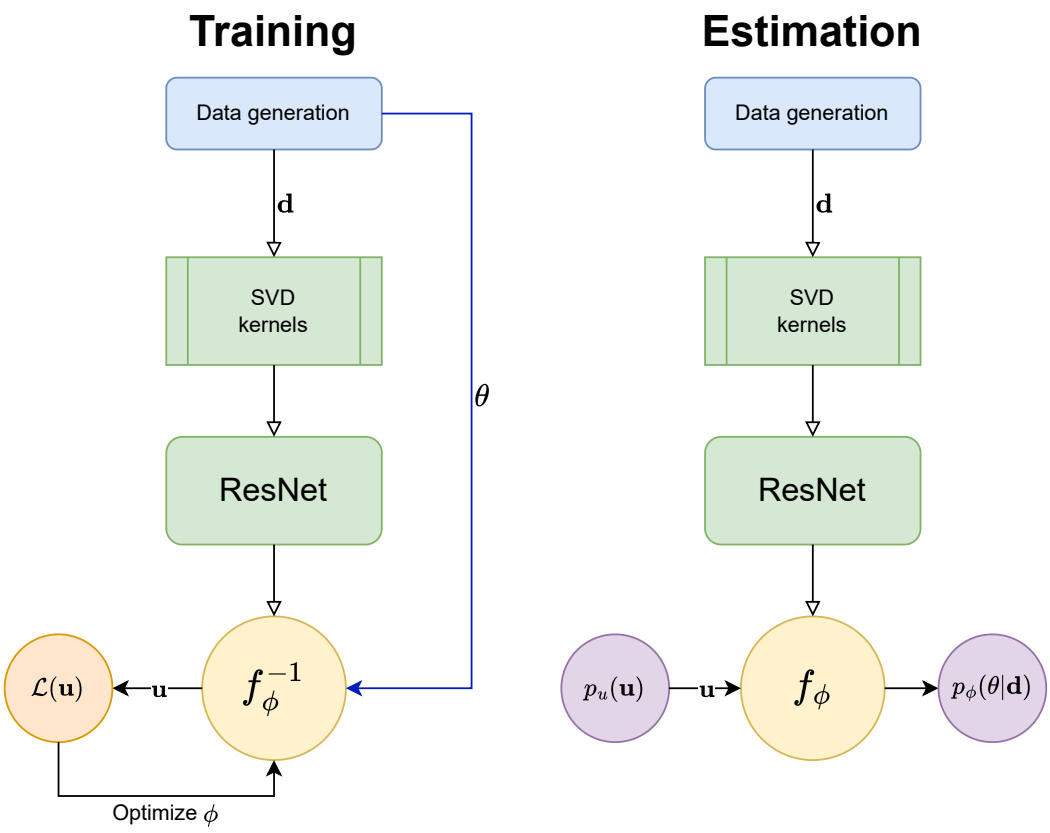


Figure 17: Network structure and operation for the training and estimation sequences.

4 Methodology

In this section, we give the details on the structure, data generation and training of our conditional normalizing flows network.

4.1 Framework implementation

The parameter estimation is handled by a conditional NF network consisting of permuted coupling layers to ensure a high training and inference speed. Fig.(17) shows an illustration detailing the structure and information flow in the training and estimation sequences of the network. The network will be trained to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathbf{d})$ where $\boldsymbol{\theta} = \{\mathcal{M}_c, q, \phi, \theta, d_L, \theta_{JN}\}$, containing the chirp mass, mass ratio, sky angles, luminosity distance and inclination angle. The most important ones for this work are the sky angles, but the other parameters can also provide information that is useful for astronomers searching for a EM companion. For example, knowing information about the masses helps with estimations

of what objects emit the GWs, and knowing the inclination angle gives information about what kind of EM companion to expect. Having the luminosity distance in addition to the sky angles helps localize the signal to find the source galaxy.

Because the input size of the signal depends on the cut frequency used, we train a diffent network at each cut frequency. The cut frequencies we train a network for are $f_{\text{cut}} \in \{40, 70, 100\}$ Hz. The higher cut frequency networks require a larger amount of SVD kernels to be used, since the frequency range seen by that network is larger. The rest of the network structure is the same for each network.

The network consists of two parts: a context network and a conditional NF network. The context network uses SVD kernels to extract important features from the data, which then get processed by a residual network to reduce dimensionality and get a more efficient representation of the data. This is then passed to the conditional NF to estimate the posterior of the signal.

To use the network, we train it on simulated BNS inspirals generated from the prior function and data generation. To do posterior estimation, we invert the flow network and condition it on the signal we want to estimate. Using the conditioned flow network, we then transform samples from a normal distribution into the estimated posterior distribution. This is illustrated in Fig.(17).

We implement the neural network framework using the `JAX` python package [119]. This package allows for automatic parallelization, using `vmap`, and just-in-time (`jit`) compilation of functions to speed up the training and sampling process compared to other neural network frameworks. We use the `equinox` [120] and `flowjax` [121] packages to implement the neural network structures used in this work.

4.2 Data generation

The data generation is divided in two parts: generating parameters from the priors and using these parameters to generate the data. In this work we use the LIGO Hanford, LIGO Livingston and Virgo detectors at design sensitivity [6, 7].

4.2.1 Priors

To generate the waveform, we draw the signal parameters from the priors shown in table 1. To make the sampling more efficient, we generate a batch of 1,000 signals every time we take a sample from the prior. These batches are then used to generate signal batches used in training.

The conditional prior used for $\mathcal{M}_c$ and q is a uniform prior with the condition that the component masses cannot be lower than $0.85M_\odot$. The PISNR is drawn from a Beta distribution. The luminosity distance is then scaled to match the PISNR. Since the PISNR determines how well-detectable the signal is, we use it to do *curriculum learning*, which makes it easier for the network to train on more difficult signals. In section 4.3.2 we go into more detail about this.

Parameter	Prior type	Minimum	Maximum
Chirp mass $\mathcal{M}_c$	Condition on minimum mass	$0.75M_\odot$	$2M_\odot$
Mass ratio q	Condition on minimum mass	0.125	1
Sky angle θ	Cosine	$-\pi/2$	$\pi/2$
Sky angle ϕ	Uniform	0	2π
Spin magnitude χ_1	Uniform	0.0	0.5
Spin magnitude χ_2	Uniform	0.0	0.5
Coalescence time t_c	Uniform	$-0.05s$	$+0.05s$
Phase Φ	Uniform	0	2π
Inclination angle θ_{JN}	Sine	0	π
Polarization angle ψ	Uniform	0	π
PISNR	Beta	5	50

Table 1: Table summarizing the prior distributions used to generate signals.

4.2.2 Data generation process

After we generate a batch of signal parameters, we generate the corresponding signals using a data generator. This process generates signals in parallel for the whole batch, and is fully coded in the `JAX` framework to make use of `jit` compiling. Because we are restricted to waveforms that have been coded in `JAX`, we are restricted to frequency domain waveforms. If we want to make sure we only use a pre-merger signal, we cannot perform Fourier transforms because location of the merger time then depends on how long we choose the signal to be. Thus, we decided to stay in frequency domain and use data up to a certain maximum frequency to make sure we only use pre-merger data. The data generation is divided in the following steps:

1. We generate the polarizations, h_+ and $h_\times$ using the source parameters and the waveform generator. We use `IMRPhenomD_NRTidalV2` [52], as implemented in the `Ripple` package [53]. The waveform is generated in the frequency domain with a minimum frequency of 20 Hz and a maximum frequency of 256 Hz.

2. Next, we project the generated waveform polarizations into the detector frame and calculate the detector response, depending on the signal parameters.

3. The detector strain is then scaled with the PISNR of the signal and whitened with the detector amplitude spectral density (ASD). The PISNR scaling is done by masking out the frequencies above the cut frequency used by the network, calculating the PISNR and then rescaling the masked frequency strain to have the correct PISNR. We use the Advanced Virgo and Advanced LIGO ASD curves from `Bilby` [74] to whiten the signal.

4. We then add Gaussian noise to the complete frequency strains: since we already whitened the strains we add whitened noise to the strains. The strains are now ready to be used.

The data generator creates frequency strain data containing the waveform together with noise up to the specified cut frequency, after which it only contains noise. This allows us to use the

same data dimensions for each network. The noise above the cut frequency is automatically filtered out by the SVD kernels, because they contain only zeros above the cut frequency, so the inner product with the SVD kernels and the noise above the cut frequency is zero.

4.3 Training the network

The network is trained in a training loop. Training a network takes about one week, with the network being trained on around 2,000,000,000 unique signals. To make it easier to train the network, we make use of curriculum learning.

4.3.1 Training loop

The training loop consists of a training and validation step, which we describe shortly. Each iteration of the training and validation loops is called an *epoch*. During each epoch, the network sees 10 batches of 1000 signals. The network is trained on 8 batches and validated on 2 other batches during the curriculum learning process. Once we reached the desired PISNR distribution, we train the network with 15 training batches and 5 validation batches in each epoch to increase the training efficiency.

We start the training loop by generating a batch of data with the data generator. The batch is then used to do a *training step*; this involves passing the data through the network, calculating the loss, and training the network by back propagation of the loss to optimize the network parameters to minimize the loss. This is repeated for 8 batches.

In the validation loop, we again generate a batch of unseen data and calculate the loss of the network, but we do not train the network with back propagation. The validation loop is used to see the network performance on unseen data, and see if that has improved since the last training loop. If the validation loss is lower than the previous validation loss, we use the new network parameters. Otherwise we continue with the parameters with the lowest validation loss.

We stop the training when we reach the maximum amount of epochs (100,000), or when we the loss does not decrease over a certain amount of epochs, called the *max patience*. In practice the network usually reaches the maximum patience before reaching the maximum amount of epochs.

4.3.2 Curriculum learning

To train the network more efficiently, we make use of a training scheme called *curriculum learning*. The idea behind this is to start the training with easier data, and ramp up the difficulty once the network has converged.

The signal's PISNR is the main parameter that determines how well-detectable a signal is. Therefore, to put curriculum learning into practice, we start training the network on signals with a higher PISNR and lower it over time. We keep the minimum and maximum of the distributions constant during the first steps of the curriculum learning process, with a minimum of 10 and a maximum of 50, and change the temperature and peak of the distribution to focus the network on a certain region in the distribution. The curriculum learning process can be summarized in the following steps:

- We start the training process by drawing the PISNR samples from a beta distribution with a temperature of 35 and a peak of 40.
- Once we reach the maximum patience or of the network, we change the temperature to 25 and the peak to 30.
- When the loss stabilizes again, we lower the temperature to 15 and peak to 20.
- In the final step of the curriculum learning process, we lower the minimum PISNR to 5, the temperature to 10 and the peak to 15. Then, we let the network train without the maximum patience for about 1.000.000 epochs, where we also increase the number of training batches to 15 and number of validation batches to 5. This is done to allow the network to fully converge to the desired accuracy.

During this process, we set the maximum patience to 500 epochs. After the network has been trained to use the desired PISNR distribution, we can continue training it for the desired number of epochs. Fig.(18) shows the PISRNs used in the curriculum learning process.

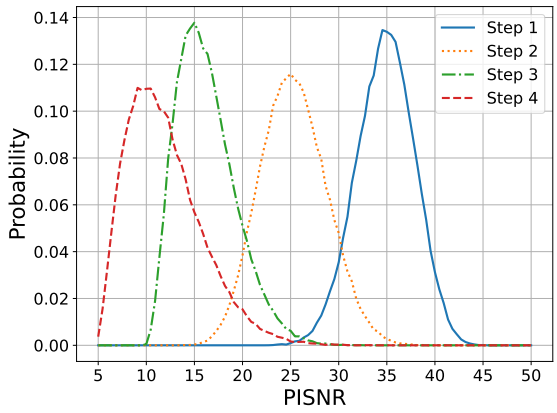


Figure 18: Normalized samples from the PISNR distributions used in this work. The figure shows the PISNR distribution steps used in the curriculum learning.

4.3.3 Network dimensions

The networks use a different number of SVD kernels depending on the cut frequency. The number of SVD kernels used are 500, 850 and 950 for the 40, 70 and 100 Hz cut frequency networks respectively. The ResNet consists of 5 blocks, each with a width of 1024. This network condenses the input signal to a dimension of 256, which is then passed to the flow network. The later consists of 4 flow layers, each with a width of 50 and a depth of 1, transformed by Bernstein polynomials of order 64.

The waveform generator generates 1000 signals with a 20481 samples in frequency space, with a minimum of 20 Hz and a maximum of 256 Hz. Initially, the samples contain the full signal, but when we take the inner product of the signals with the SVD kernels, the signal will be masked with the correct cut frequency, depending on which network we use. This means that we can pass the exact same signal to the different networks to compare them without any additional computations.

5 Results

Now that we discussed the construction and operation of our network, we move on to discussing the results. First we discuss the accuracy of the network. Then we do a more detailed investigation of posteriors produced by the network, before moving on to produce posteriors for events with parameters like those that have already been observed, such as GW170817 and GW190425.

5.1 Network accuracy

First of all, we want to investigate the accuracy of the trained networks. The first step is to look at the posterior estimations generated by the networks resulting from the analysis of a signal. Fig.(19) shows an example of a signal analysed by the 40, 70 and 100 Hz networks, with a fixed PISNR. This is done to provide a fair comparison and avoid the broadening of the posteriors due to total PISNR of the same signal decreasing when we lower the maximum frequency. Nevertheless, we observe that the recovered posteriors are broader for a lower cut frequency network. This is expected, as these networks see less cycles in their sensitivity band and have less information to estimate the signal. The posteriors estimated by the networks properly recover the true values for the sky location and the other parameters relevant for EM follow-up.

To further verify the reliability of the networks, we investigate if there is bias present in the estimations done by the networks. To do this, we generate probability-probability plots (PP-plots). These show the fraction of injections for which the true value falls within a given confidence interval. If the posterior estimations show no bias, the lines representing the different parameters align along the diagonal. Fig.(24) shows the PP-plots generated for the 40 and 100 Hz networks. The p-values next to the parameters measure the probability of obtaining the observed results for the parameter. These results do not guarantee that the networks are unbiased, but it is a necessary condition for the networks to be trustworthy. With these tests we can conclude that the networks have the expected statistical behaviour for a large number of events, and we can continue with a more detailed analysis of the network performance.

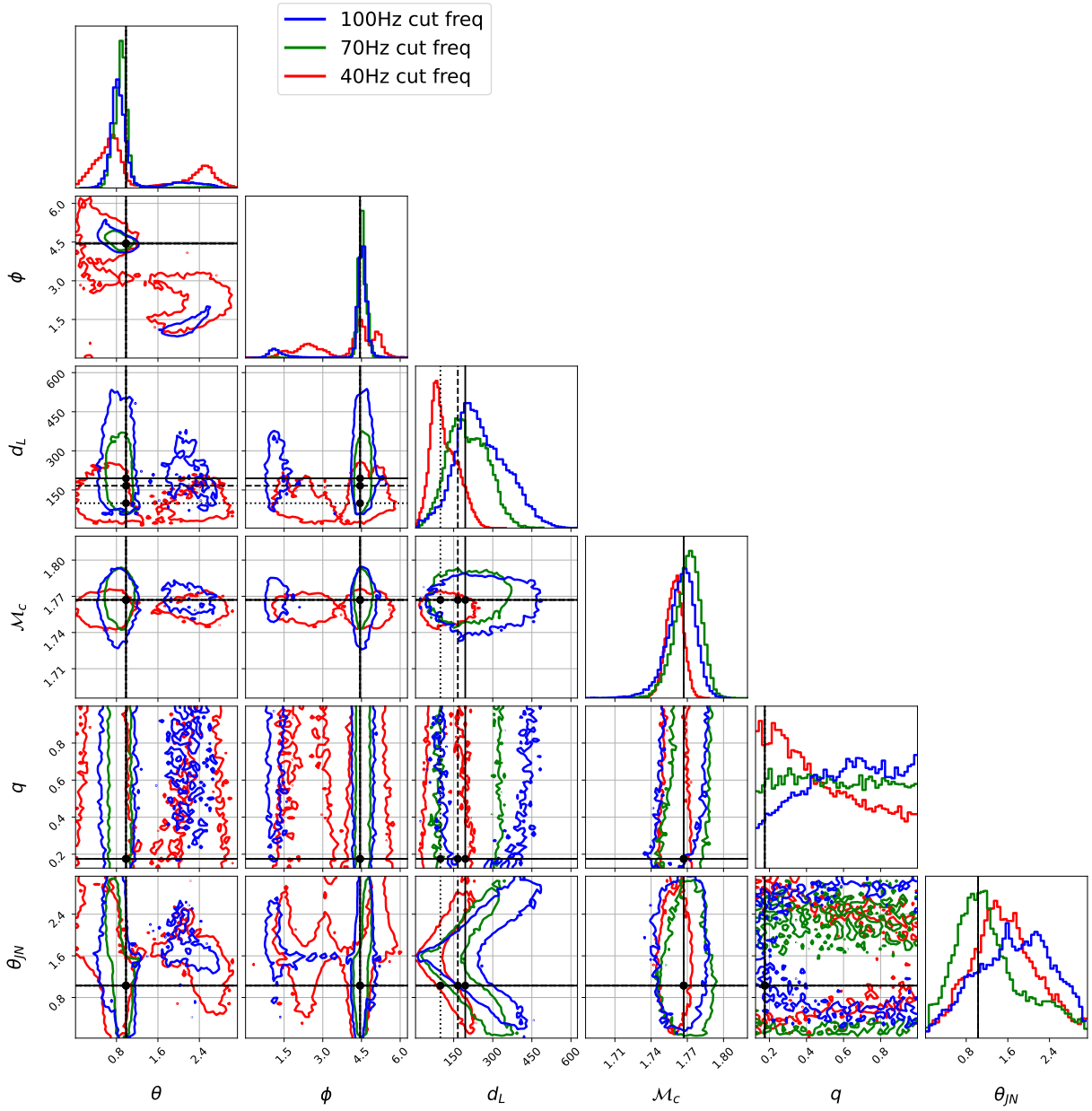


Figure 19: Corner plot representing the posterior estimations generated by the networks for a inspiraling BNS signal, containing the posterior estimations for the 40, 70 and 100Hz networks. The contours contain the 90% confidence intervals. To compare the networks fairly we scale the signals seen by the networks such that they all have a constant PISNR of 13.5. This means that the signals seen by the networks have a different luminosity distance: the dotted line, the dashed line and the solid line indicate the true value for the luminosity distance for the 40, 70 and 100 Hz networks respectively. For all the maximum frequencies the networks recover the posterior correctly, but the posteriors become broader for the lower frequency networks. This is expected, because the lower frequency networks observe less cycles of the signal in their sensitivity band.

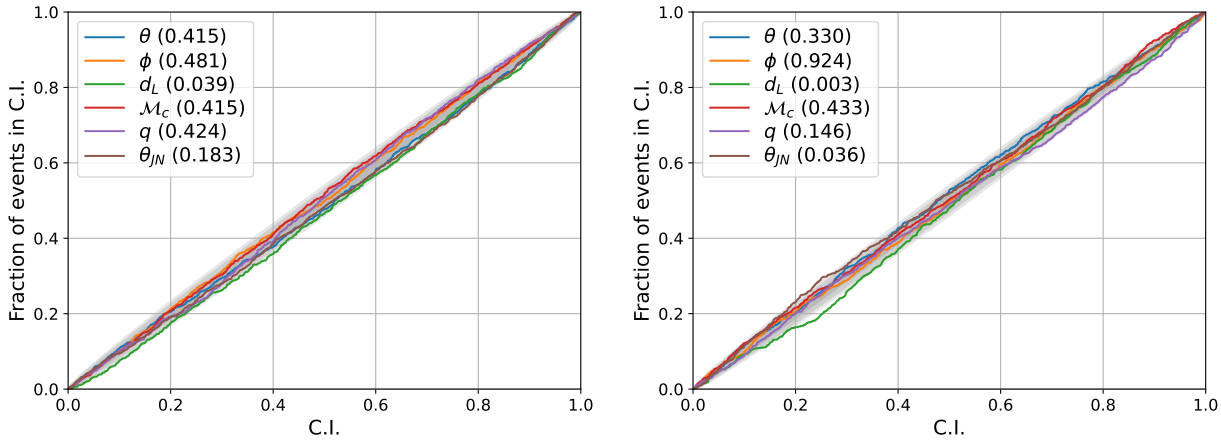


Figure 20: PP plots for the 40Hz network (left) and the 100Hz network (right). The numbers behind the parameters are the p-values of that number. The grey area around the diagonal is the 99% confidence interval. The lines closely follow the diagonal, suggesting that the networks are unbiased.

5.2 Investigation of the inference results

To investigate the network results further, we focus on the sky locations estimated by the networks. Fig.(21) shows evolving skymaps for two situations: one with a constant PISNR for all the detectors, and one with a constant luminosity distance, mimicking a real detection.

In the left figure, all parameters are the same except the luminosity distances, since the PISNRs are kept constant across networks. This again shows that the performance of the high cut frequency networks is better due to the signal having more cycles in their sensitivity band, improving observations of the dephasing and time delay difference which allows the network to better reconstruct the origin of the signal. The improvement is the same for other recovered parameters, except for the chirp mass. It seems the quality of the chirp mass estimation mainly depends on the relative amplitude of the noise and the signal, which is smaller when in the low cut frequency networks to maintain a constant PISNR.

The right figure shows the sky estimation of a signal with a constant luminosity distance across the networks, corresponding to a more realistic scenario. The sky map evolves as expected, with the largest estimated sky area for the lowest maximum frequency. Here, the difference in performance is bigger since the low cut frequency networks also have a lower PISNR. Due to this, the earliest obtained skymaps may not be good enough to accurately locate the host galaxy before the merger in some of the scenarios. This has also been observed in previous studies [29, 30], so this effect is to be expected. The other estimated parameters can produce usable results, even if the skymap is not good, see for example Fig.(26).

For the constant luminosity distance estimation, the total SNR of the signal needs to be around 30 to be loud enough to have the PISNRs of 8.9, 14.6 and 17.1. Such loud BNS signals have only been observed once, with the GW170817 detection, but are expected to increase in detection rate with coming detector upgrades. This also increases the rate of detection for high PISNR signals: for example, if we were to observe GW170817 with upgraded detectors, the SNR would be around 80, which is more than enough to produce a skymap usable for

EM follow-up.

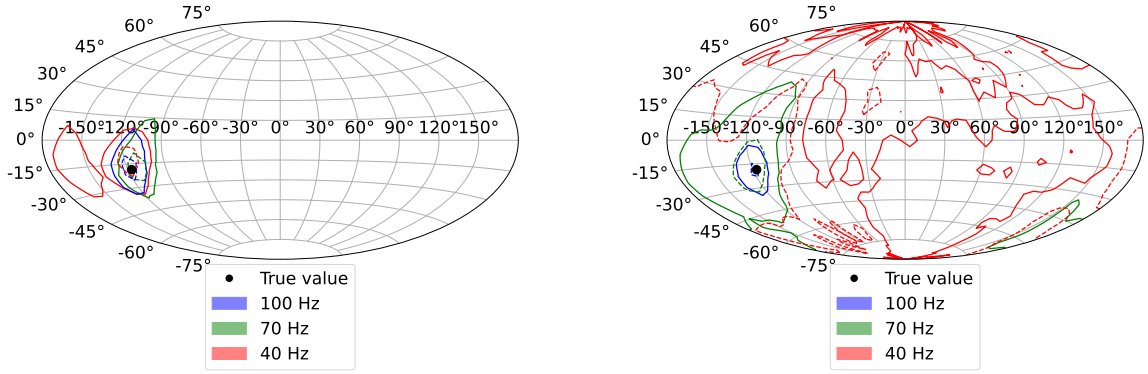


Figure 21: Evolving skymaps estimated by the networks. The darker areas are the 90% confidence intervals, and the lighter areas are the 90% confidence intervals. The left figure is generated with a constant PISNR of 17.1 across the networks. This shows that the sky area estimated by the lower frequency networks is larger than the sky area estimated by the higher frequency networks. On the right, we have a more realistic scenario; these sky locations were estimated at a constant luminosity distance across the networks. The PISNRs in the networks are 8.9, 14.6 and 17.1. In this case, the difference between the networks is larger since the lower frequency networks also have a lower PISNR.

To further investigate the accuracy of the skymaps generated by the networks, we calculated skymap area estimated by the networks for 1000 samples using the `HEALPix`[15] skymap estimation software [122]. The cumulative density plots obtained from there results are shown in Fig.(22). These figures confirm the expected behaviour; overall, the high cut frequency networks perform better than the low cut frequency networks. In the upper right of the plot, the 40and 70 Hz networks seem to outperform the 100 Hz network. This can indicate that the 100 Hz network requires more training time, since the 40 and 70 Hz networks showed this behaviour too when they were less well-trained. The results shown are similar to those obtained by previous studies [29, 30][16], but we want to stress that our approach does not rely on matched filtering and can estimate posteriors for other parameters relevant for the EM follow-up.

The sky areas estimated are regularly good enough to enable follow-up studies: for example, the GRANDMA telescope network [123], which has already done follow-up studies in the third LIGO-Virgo observing run, had a average coverage of 183 deg^2 per observation. The

[15] `HEALPix` website: `http://healpix.sourceforge.net`

[16] The PISNR distributions used in this study and the previous studies are not the same, so this makes a quantative comparison harder.

100 Hz network estimations of the 50% confidence interval is below this limit in about 40% of the estimated signals, while the 90% confidence interval estimations reach this in about 10% of the estimated sky areas for the same network. Depending on the type of follow-up study, the coverage of the telescopes can be larger. For example, the Fermi Gamma-Ray Space Telescope's Large Area Telescope [124] has a coverage of about 25920 deg^2. Most of the 90% confidence intervals estimated by the networks fall below this value.

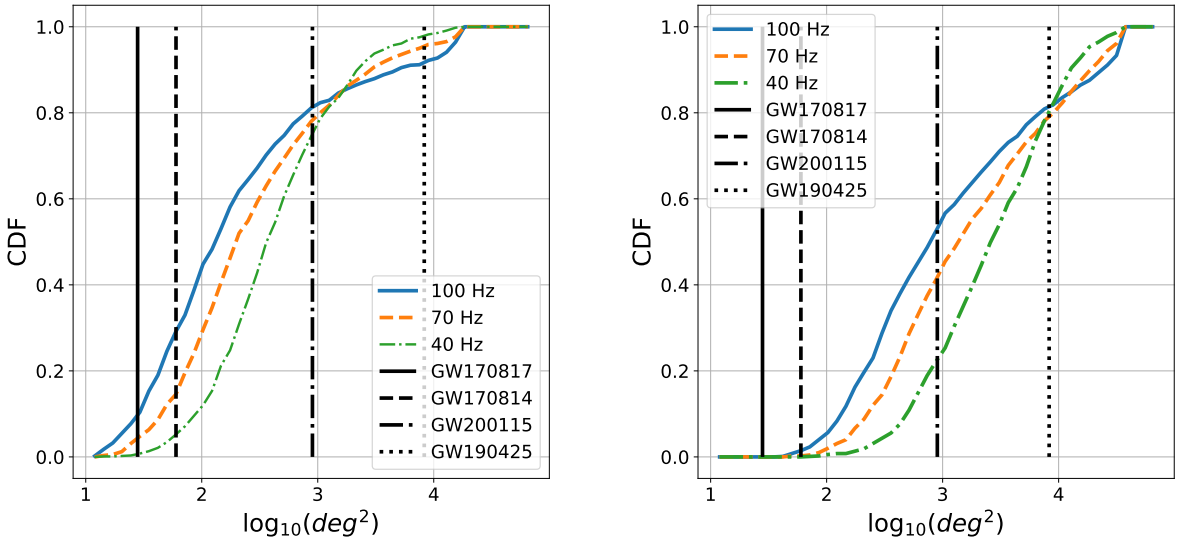


Figure 22: Cumulative density functions for the log of the estimated sky areas in degrees squared. The left figure shows the area of the 50% confidence interval, while the figure on the right shows the area of the 90% confidence interval. As observed earlier, the general trend is that the high cut frequency networks produce better results than the low cut frequency networks, in most of the cases. This is discussed further in the discussion in section 6.1.2. As a reference, we included the 90% confidence interval sky areas obtained for a few GW observations: GW170817 [11] and GW190425 [125] are two BNS observations, GW200115 [126] is a NSBH merger, and GW170814 [127] was the first BBH merger observed with all 3 detectors active. Note that these results were obtained with parameter estimation after the full signal was observed.

5.3 Analysis of realistic events

To further evaluate the network performance, we analyse realistic events in the pre-merger phase. In particular, we analyse GW170817 [11] and GW190425 [125] like events. We take the median values of the parameters reported in the observation papers, and inject those values into noise generated using the detector ASDs used in this work. Fig.(23) shows the resulting skymaps. We use all three detectors, while GW190425 was observed by only the two LIGO detectors. For the GW170817-like event, we scaled the luminosity distance such that we obtain a total SNR which is similar to the observed SNR of GW170817. The GW190425 event does not use a scaled luminosity distance. The full corner plots can be found in the

appendix, in Fig.(25) and Fig.(26).

For the GW170817-like event, the 40 and 70Hz max frequencies perform poorly, but the 50% confidence interval of the latter is reasonably well constrained. The 100Hz sky location is good enough for EM follow-up. However, this is fairly late in the merger, so this might not result in pre-merger localization, but it can facilitate a fast follow-up.

The GW190425-like sky estimations are better in the 70Hz frequency range, which is more likely to result pre-merger EM follow-up observations.

Thus, we have shown that the framework can produce skymaps useful for EM follow-up with reasonable signals without relying on matched filtering, with the required detector upgrades.

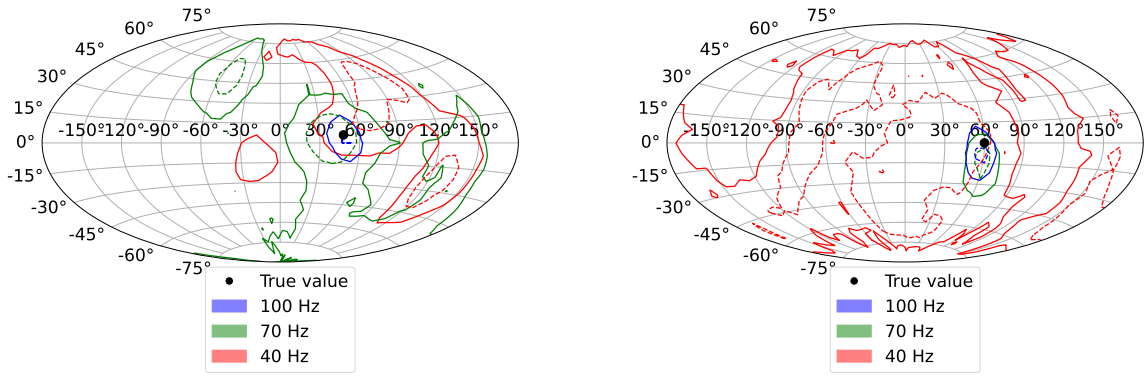


Figure 23: Skymaps generated for realistic events. The darker areas are the 50% confidence intervals, and the lighter areas are the 90% confidence intervals. The left figure shows the skymap for a GW170817-like event, with comparable sky position, source parameters and observed SNR. The PISNRs in this case are 15.0, 19.2 and 22.5 for the 40, 70 and 100Hz networks respectively. The 40Hz skymap does not allow for EM follow-up, but the 50% confidence interval for 70Hz is usable, and the 100Hz skymap is reasonably good. The right figure shows the results for the GW190425-like injection. The PISNRs are 10.0, 16.7 and 19.6 for the 40, 70 and 100Hz maximum frequencies respectively. Again, the 40Hz results are not great, but the 70 and 100Hz networks recover a usable sky location.

6 Discussion, conclusion and outlook

6.1 Discussion

In this work, we developed a conditional normalizing flows framework capable of pre-merger sky localization of BNS signals, in addition to estimating other parameters relevant for EM follow-up. We train the framework to estimate the sky location, component masses, luminosity distance and inclination angle of pre-merger BNS inspiral signals with a minimum frequency of 20 Hz and a constant cut frequency. The framework produces posteriors for the sky angles, chirp mass, mass ratio, luminosity distance and inclination angle. These parameters are the most relevant to EM follow-up observations, because they give information about the probability of observing follow-up signals, but other parameters could also be estimated by increasing the network expressivity. The networks were trained with a maximum frequency of 40, 70 and 100 Hz. We investigated the performance by examining the posteriors estimated by the networks and confirmed that the networks have no bias by investigating PP plots produced by the networks. Then we closely examined the sky localizations produced by the network for simulated events and events that have already been observed.

6.1.1 Network design and performance

The maximum frequencies used in this work were chosen as a proof-of-concept, and can be chosen to have a more continuous frequency range to create a more continuous evolving skymap. The maximum frequencies chosen in this work are relatively high because enough cycles of the signal need to be observed to produce a usable skymap. For some signals, the PISNRs obtained for the low frequency networks are not high enough to produce usable sky estimations, note however that other parameters can have a reasonable posterior estimation. Even if the sky location is not known exactly, it can be useful to know if the inclination angle of a merger is face-on, which increases the likelihood of observing a GRB, since those are emitted perpendicular to the orbital plane [128]. Knowing the luminosity distance also lets us estimate if the merger is close enough ($d_L < 80$ Mpc) to produce an observable GRB [128].

The exploratory nature of this research required a quick iteration of network design, which resulted in less time being available to let the networks converge properly by training them longer. In principle, the results could be improved upon by enhancing the training scheme and training the networks for more epochs. Further improvements can be done by optimizing the hyperparameters such as the batch size, number of batches in a epoch, and testing different network dimensions to find the best-performing network. Such a hyperparameter optimization is very time-consuming but a good way to optimize the performance.

6.1.2 Area cumulative density functions

The cumulative density functions of the area estimations in Fig.(22) show a peculiar feature: on top right of the figure, the performance of the 70 Hz and 40 Hz network seems to be better then the performance of the 100 Hz network, because the 100 Hz network dips below the other networks. We think this means that the 100 Hz network has not trained enough, because the 40 Hz and 70 Hz network had these same features in a earlier stage of the training. This

problem could be mediated by training the network for longer, but due to the limited time frame of this work, we leave this to future investigations.

6.1.3 Detector sensitivity

The networks were trained with the current detector low frequency cutoff of 20 Hz. Since BNS signals spend a lot of time (up to 1.5 hours) in the 5-20 Hz frequency range, the performance of the networks could be improved further by lowering the used minimum frequency. This would also drastically increase the size of the strain that the network needs to analyse, but this can be mitigated by using adaptive frequency resolution [129] or relative binning [130] to reduce the size of the input data. The current detectors are not sensitive enough to produce usable data in the 5-20 Hz frequency range, but planned detectors like the Einstein telescope or Cosmic Explorer will have a better sensitivity in the low frequency range. If we consider signals with a minimum frequency of around 5 Hz, it would be critical to account for the Earth's rotation due to the length of the observed signals. This could also improve the localization capabilities, since the modulation of a signal with the Earth's rotation provide information about the signal's source location [131].

6.1.4 Framework comparison

To compare our framework to the `BAYESTAR` [29] and `GW-SkyLocator` [30] frameworks, we consider a number of differences. The main differences between our network and the previous frameworks are as follows:

1. The network developed in this work does not rely on matched-filtering based approaches.

2. Our framework can provide estimations of intrinsic parameters in addition to the sky location and luminosity distance.

3. The framework can provide parameter estimations for signals where the detector PISNR is below 4 in at least a single detector. The previous frameworks disregard these signals their analysis.

Due to the differences with previous frameworks, our framework could be expanded to include better waveform models to increase the accuracy with respect to real data and estimate additional intrinsic parameters such as the neutron star spins.

6.1.5 Sample leakage for chirp mass

In some corner plots, the posterior estimations for the chirp mass seem to be very peaked around the estimated value. This is a issue with plotting, but also a sign of sample leakage: the bins are equally distributed between the minimum and maximum of the samples, so if a estimation has a few samples far away from the bulk of the estimated samples, the bins will be very wide, and the bulk of the samples will fall into a few bins, such as the 70 Hz network in Fig.(25) and the 40 Hz network in Fig.(26). This issue can indicate that the network has not converged fully yet, and could be solved with more training time.

6.2 Conclusion

This work presents conditional normalizing flows neural networks capable of estimating the sky location, the component masses, luminosity distance and inclination angle for EM-follow up from pre-merger BNS inspirals. The sky localizations by the network are regularly good enough for pre-merger follow-up observations. While more work needs to be done to develop the framework into a complete detection pipeline, it presents an interesting avenue for future pre-merger parameter estimation frameworks for inspiraling signals with EM counterparts.

6.3 Outlook

6.3.1 Non-Gaussian noise

To investigate the robustness of the network to non-Gaussian noise, we can investigate the performance if we inject irregular noise, such as glitches. Non-Gaussian noise bursts can be very detrimental to matched filtering based approaches, since they can have similar shape to parts of the inspiral, which can then provide high SNR triggers on mismatched templates. These can be generated [132] and injected into the strain to see how resilient the network is with respect to non-Gaussian noise bursts. If this is successful, it would give this approach an additional edge over previous studies.

If we want the framework to be usable in real data, we also need to consider the effect of variations in the detector PSDs. These shifts can lead to variations in detector sensitivities which could affect the performance of the network. This problem has been investigated before by the `DINGO` framework [33], where they also trained the network on sampled PSD distributions. This could be implemented in our framework as well to solve this issue.

6.3.2 Detection pipeline

One interesting future opportunity of the framework developed in this work is to adapt it into a single-framework detection pipeline. One could use the output of the trained context network as input for a binary classification neural network [133], which could provide triggers when a signal is present in the data. This would enable the framework to be a single framework detection and characterization pipeline, as far as we know the first of its kind. This would aid in reducing the latency further since it would require less communication with other pipelines.

7 Laymen summary

Zwaartekrachtsgolven zijn kleine trillingen in de ruimtetijd die worden gegenereerd door extreme-massa systemen, zoals binaire neutronensterren die gaan fuseren. Zo'n systeem van binaire neutronensterren is al geobserveerd in de GW170817 detectie. Bij deze detectie hebben we de kans gekregen om veel interessante observaties te doen, maar we hebben wel 10 uur aan data verloren omdat de locatie pas een paar uur na de detectie bekend was. Om meer interessante dingen te weten te komen, willen we graag *voor* de fusering van de sterren de bron localizeren. In dit werk hebben we een netwerk ontwikkeld dat binnen een paar seconden de locatie van de bron kan bepalen. Ook kan het netwerk de intrinsieke parameters

van het signaal inschatten, zoals de massas van de neutronensterren, zonder dat het netwerk afhankelijk is van andere detectie algoritmes zoals matched filtering.

8 References

References

[1] A. Einstein, Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften pp. 688–696 (1916).

[2] A. Einstein, Annalen der Physik **354**, 769 (1916), https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.19163540702, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19163540702.

[3] K. Riles, Prog. Part. Nucl. Phys. **68**, 1 (2013), 1209.0667.

[4] R. A. Hulse and J. H. Taylor, Astrophys. J., Lett., v. 195, no. 2, pp. L51-L53 (1975), URL https://www.osti.gov/biblio/4215694.

[5] B. P. Abbott et al. (LIGO Scientific, Virgo), Phys. Rev. Lett. **116**, 241102 (2016), 1602.03840.

[6] J. Aasi, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, et al., Classical and Quantum Gravity **32**, 074001 (2015), URL https://doi.org/10.1088/0264-9381/32/7/074001.

[7] F. Acernese, M. Agathos, K. Agatsuma, D. Aisa, N. Allemandou, A. Allocca, J. Amarni, P. Astone, G. Balestri, G. Ballardin, et al., Classical and Quantum Gravity **32**, 024001 (2014), URL https://doi.org/10.1088/0264-9381/32/2/024001.

[8] F. S. Broekgaarden, S. Banagiri, and E. Payne (2023), 2303.17628.

[9] M. Maggiore, C. V. D. Broeck, N. Bartolo, E. Belgacem, D. Bertacca, M. A. Bizouard, M. Branchesi, S. Clesse, S. Foffa, J. Garcí a-Bellido, et al., Journal of Cosmology and Astroparticle Physics **2020**, 050 (2020), URL https://doi.org/10.1088%2F1475-7516%2F2020%2F03%2F050.

[10] E. D. Hall, K. Kuns, J. R. Smith, Y. Bai, C. Wipf, S. Biscans, R. X. Adhikari, K. Arai, S. Ballmer, L. Barsotti, et al., Physical Review D **103** (2021), URL https://doi.org/10.1103%2Fphysrevd.103.122004.

[11] B. P. Abbott et al. (LIGO Scientific, Virgo), Phys. Rev. Lett. **119**, 161101 (2017), 1710.05832.

[12] C. Meegan, G. Lichti, P. N. Bhat, E. Bissaldi, M. S. Briggs, V. Connaughton, R. Diehl, G. Fishman, J. Greiner, A. S. Hoover, et al., The Astrophysical Journal **702**, 791–804 (2009), ISSN 1538-4357, URL http://dx.doi.org/10.1088/0004-637X/702/1/791.

[13] V. Savchenko, C. Ferrigno, E. Kuulkers, A. Bazzano, E. Bozzo, S. Brandt, J. Chenevez, T. J. L. Courvoisier, R. Diehl, A. Domingo, et al., Atrophys. Journal. Lett. **848**, L15 (2017), 1710.05449.

[14] B. P. Abbott et al. (LIGO Scientific, Virgo, Fermi-GBM, INTEGRAL), Astrophys. J. Lett. **848**, L13 (2017), 1710.05834.

[15] P. Mészáros, D. B. Fox, C. Hanna, and K. Murase, Nature Rev. Phys. **1**, 585 (2019), 1906.10212.

[16] B. P. Abbott et al. (LIGO Scientific, Virgo, 1M2H, Dark Energy Camera GW-E, DES, DLT40, Las Cumbres Observatory, VINROUGE, MASTER), Nature **551**, 85 (2017), 1710.05835.

[17] B. P. Abbott et al. (LIGO Scientific, Virgo), Phys. Rev. Lett. **121**, 161101 (2018), 1805.11581.

[18] A. Nishizawa, Phys. Rev. D **93**, 124036 (2016), 1601.01072.

[19] E. R. Most and A. A. Philippov, The Astrophysical Journal Letters **893**, L6 (2020), URL https://dx.doi.org/10.3847/2041-8213/ab8196.

[20] M. Nicholl, E. Berger, D. Kasen, B. D. Metzger, J. Elias, C. Briceño, K. D. Alexander, P. K. Blanchard, R. Chornock, P. S. Cowperthwaite, et al., The Astrophysical Journal Letters **848**, L18 (2017), URL https://dx.doi.org/10.3847/2041-8213/aa9029.

[21] N. Bucciantini, B. D. Metzger, T. A. Thompson, and E. Quataert, Monthly Notices of the Royal Astronomical Society **419**, 1537 (2011), ISSN 0035-8711, https://academic.oup.com/mnras/article-pdf/419/2/1537/3125386/mnras0419-1537.pdf, URL https://doi.org/10.1111/j.1365-2966.2011.19810.x.

[22] S. Sachdev et al., Astrophys. J. Lett. **905**, L25 (2020), 2008.04288.

[23] Q. Chu et al., Phys. Rev. D **105**, 024023 (2022), 2011.06787.

[24] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Class. Quant. Grav. **33**, 175012 (2016), 1512.02864.

[25] G. Baltus, J. Janquart, M. Lopez, H. Narola, and J.-R. Cudell, Phys. Rev. D **106**, 042002 (2022), 2205.04750.

[26] H. Yu, R. X. Adhikari, R. Magee, S. Sachdev, and Y. Chen, Phys. Rev. D **104**, 062004 (2021), 2104.09438.

[27] J. Veitch et al., Phys. Rev. D **91**, 042003 (2015), 1409.7215.

[28] J. Skilling, Bayesian Analysis **1**, 833 (2006).

[29] L. P. Singer and L. R. Price, Phys. Rev. D **93**, 024013 (2016), 1508.03634.

[30] C. Chatterjee and L. Wen, The Astrophysical Journal **959**, 76 (2023).

[31] A. Kolmus, G. Baltus, J. Janquart, T. van Laarhoven, S. Caudill, and T. Heskes, Phys. Rev. D **106**, 023032 (2022), 2111.00833.

[32] J. Langendorff, A. Kolmus, J. Janquart, and C. Van Den Broeck, Phys. Rev. Lett. **130**, 171402 (2023), 2211.15097.

[33] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Phys. Rev. Lett. **127**, 241103 (2021), 2106.12594.

[34] M. Dax, S. R. Green, J. Gair, M. Pürrer, J. Wildberger, J. H. Macke, A. Buonanno, and B. Schölkopf, Phys. Rev. Lett. **130**, 171403 (2023), 2210.05686.

[35] T. Wouters, P. T. H. Pang, T. Dietrich, and C. Van Den Broeck (2024), 2404.11397.

[36] J. M. Antelis, J. M. Hernández, and C. Moreno, J. Phys. Conf. Ser. **1030**, 012005 (2018).

[37] S. C. Chris van den Broek, *Gravitational waves* (2021).

[38] S. Chandrasekhar, The Astrophysical Journal **142**, 1488 (1965).

[39] L. Barack, Class. Quant. Grav. **26**, 213001 (2009), 0908.1664.

[40] M. Favata/SXS/K. Thorne, *Sounds of spacetime* (2017), URL https://www.soundsofspacetime.org/the-basics-of-binary-coalescence.html.

[41] H. Tagoshi and M. Sasaki, Prog. Theor. Phys. **92**, 745 (1994), gr-qc/9405062.

[42] C. Palenzuela, Frontiers in Astronomy and Space Sciences **7** (2020), ISSN 2296-987X, URL http://dx.doi.org/10.3389/fspas.2020.00058.

[43] A. Jan, D. Ferguson, J. Lange, D. Shoemaker, and A. Zimmerman, *Accuracy limitations of existing numerical relativity waveforms on the data analysis of current and future ground-based detectors* (2023), 2312.10241.

[44] L. Blanchet, International Journal of Modern Physics D (2018), URL https://api.semanticscholar.org/CorpusID:119186690.

[45] A. Buonanno, A. Buonanno, and T. Damour, Physical Review D **59**, 084006 (1998), URL https://api.semanticscholar.org/CorpusID:14951569.

[46] P. Ajith, M. Hannam, S. Husa, Y. Chen, B. Brügmann, N. Dorband, D. Müller, F. Ohme, D. Pollney, C. Reisswig, et al., Physical Review Letters **106** (2011), ISSN 1079-7114, URL http://dx.doi.org/10.1103/PhysRevLett.106.241101.

[47] S. Husa, S. Khan, M. Hannam, M. Pürrer, F. Ohme, X. J. Forteza, and A. Bohé, Phys. Rev. D **93**, 044006 (2016), URL https://link.aps.org/doi/10.1103/PhysRevD.93.044006.

[48] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. J. Forteza, and A. Bohé, Physical Review D **93** (2016), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.93.044007.

[49] F. Messina, R. Dudi, A. Nagar, and S. Bernuzzi, Physical Review D **99** (2019), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.99.124051.

[50] T. Dietrich, S. Bernuzzi, and W. Tichy, Physical Review D **96** (2017), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.96.121501.

[51] L. Wade, J. D. E. Creighton, E. Ochsner, B. D. Lackey, B. F. Farr, T. B. Littenberg, and V. Raymond, Phys. Rev. D **89**, 103012 (2014), URL https://link.aps.org/doi/10.1103/PhysRevD.89.103012.

[52] T. Dietrich, A. Samajdar, S. Khan, N. K. Johnson-McDaniel, R. Dudi, and W. Tichy, Physical Review D **100** (2019), ISSN 2470-0029, URL http://dx.doi.org/10.1103/PhysRevD.100.044003.

[53] T. D. P. Edwards, K. W. K. Wong, K. K. H. Lam, A. Coogan, D. Foreman-Mackey, M. Isi, and A. Zimmerman (2023), 2302.05329.

[54] I. Belahcene, Ph.D. thesis (2019).

[55] A. A. Michelson and E. W. Morley, American Journal of Science **34**, 333 (1887).

[56] B. Willke et al., Class. Quant. Grav. **19**, 1377 (2002).

[57] T. Akutsu et al. (KAGRA), Nature Astron. **3**, 35 (2019), 1811.08079.

[58] M. Colpi et al. (2024), 2402.07571.

[59] J. Casanueva Diaz, *Fabry-Perot Cavities in Advanced Virgo* (Springer International Publishing, Cham, 2018), pp. 37–83, ISBN 978-3-319-96014-2, URL https://doi.org/10.1007/978-3-319-96014-2_5.

[60] M. H. Phelps, K. E. Gushwa, and C. I. Torrie, in *Laser-Induced Damage in Optical Materials: 2013*, edited by G. J. Exarhos, V. E. Gruzdev, J. A. Menapace, D. Ristau, and M. Soileau (2013), vol. 8885 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, p. 88852E.

[61] M. Arain and G. Mueller, Optics express **16**, 10018 (2008).

[62] M. Granata, M. Barsuglia, R. Flaminio, A. Freise, S. Hild, and J. Marque, Journal of Physics: Conference Series **228**, 012016 (2010).

[63] B. J. Meers, Phys. Rev. D **38**, 2317 (1988), URL https://link.aps.org/doi/10.1103/PhysRevD.38.2317.

[64] *Ligo collaboration*, URL https://www.ligo.caltech.edu/page/ligos-ifo.

[65] *Virgo collaboration*, URL http://public.virgo-gw.eu/the-virgo-collaboration/.

[66] V. Varma, P. Ajith, S. Husa, J. Bustillo, M. Hannam, and M. Pürrer, Physical Review D **90** (2014).

[67] B. F. Schutz, Class. Quant. Grav. **28**, 125023 (2011), 1102.5421.

[68] P. Ajith et al., Phys. Rev. D **77**, 104017 (2008), [Erratum: Phys.Rev.D 79, 129901 (2009)], 0710.2335.

[69] Q. Chu, M. Kovalam, L. Wen, T. Slaven-Blair, J. Bosveld, Y. Chen, P. Clearwater, A. Codoreanu, Z. Du, X. Guo, et al., *The spiir online coherent pipeline to search for gravitational waves from compact binary coalescences* (2020).

[70] G. Baltus, J. Janquart, M. Lopez, H. Narola, and J.-R. Cudell, Phys. Rev. D **106**, 042002 (2022), 2205.04750.

[71] G. Baltus, J. Janquart, M. Lopez, A. Reza, S. Caudill, and J.-R. Cudell, Phys. Rev. D **103**, 102003 (2021), 2104.00594.

[72] J. Veitch et al., Phys. Rev. D **91**, 042003 (2015), 1409.7215.

[73] C. M. Biwer, C. D. Capano, S. De, M. Cabero, D. A. Brown, A. H. Nitz, and V. Raymond, Publ. Astron. Soc. Pac. **131**, 024503 (2019), 1807.10312.

[74] G. Ashton, M. Hübner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divakarla, P. J. Easter, B. Goncharov, et al., The Astrophysical Journal Supplement Series **241**, 27 (2019), URL https://doi.org/10.3847.

[75] A. Heger, C. L. Fryer, S. E. Woosley, N. Langer, and D. H. Hartmann, Astrophys. J. **591**, 288 (2003), astro-ph/0212469.

[76] J. L. Zdunik, M. Fortin, and P. Haensel, Astron. Astrophys. **599**, A119 (2017), 1611.01357.

[77] F. Douchin and P. Haensel, Astron. Astrophys. **380**, 151 (2001), astro-ph/0111092.

[78] G. F. Burgio, H. J. Schulze, I. Vidana, and J. B. Wei, Prog. Part. Nucl. Phys. **120**, 103879 (2021), 2105.03747.

[79] T. Hinderer, Astrophys. J. **677**, 1216 (2008), [Erratum: Astrophys.J. 697, 964 (2009)], 0711.2420.

[80] K. C. Gendreau, Z. Arzoumanian, and T. Okajima, in *Space Telescopes and Instrumentation 2012: Ultraviolet to Gamma Ray*, edited by T. Takahashi, S. S. Murray, and J.-W. A. den Herder (2012), vol. 8443 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, p. 844313.

[81] P. Wang, H. Yang, and X. Zhang, Phys. Lett. B **718**, 265 (2012), 1110.5550.

[82] M. Branchesi, in *Bruno Touschek 100 Years*, edited by L. Bonolis, L. Maiani, and G. Pancheri (Springer International Publishing, Cham, 2023), pp. 255–266, ISBN 978-3-031-23042-4.

[83] N. Aghanim et al. (Planck), Astron. Astrophys. **641**, A6 (2020), [Erratum: Astron.Astrophys. 652, C4 (2021)], 1807.06209.

[84] P. L. Kelly et al., Science **380**, abh1322 (2023), 2305.06367.

[85] P. Goldreich and W. H. Julian, **157**, 869 (1969).

[86] M. Lyutikov, Monthly Notices of the Royal Astronomical Society **483**, 2766 (2018), ISSN 0035-8711, https://academic.oup.com/mnras/article-pdf/483/2/2766/27201523/sty3303.pdf, URL https://doi.org/10.1093/mnras/sty3303.

[87] J. M. Lattimer and D. N. Schramm, **192**, L145 (1974).

[88] D. Eichler, M. Livio, T. Piran, and D. N. Schramm, **340**, 126 (1989).

[89] F. K. Thielemann, A. Arcones, R. Käppeli, M. Liebendörfer, T. Rauscher, C. Winteler, C. Fröhlich, I. Dillmann, T. Fischer, G. Martinez-Pinedo, et al., Progress in Particle and Nuclear Physics **66**, 346 (2011).

[90] F. Özel, D. Psaltis, S. Ransom, P. Demorest, and M. Alford, The Astrophysical Journal Letters **724**, L199 (2010), URL https://dx.doi.org/10.1088/2041-8205/724/2/L199.

[91] B. Giacomazzo and R. Perna, The Astrophysical Journal Letters **771**, L26 (2013), URL https://dx.doi.org/10.1088/2041-8205/771/2/L26.

[92] B. Kiziltan, A. Kottas, M. De Yoreo, and S. E. Thorsett, Astrophys. J. **778**, 66 (2013), 1309.6635.

[93] B. D. Metzger and A. L. Piro, Mon. Not. Roy. Astron. Soc. **439**, 3916 (2014), 1311.1519.

[94] S. Sachdev et al., Astrophys. J. Lett. **905**, L25 (2020), 2008.04288.

[95] Q. Chu et al., Phys. Rev. D **105**, 024023 (2022), 2011.06787.

[96] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Class. Quant. Grav. **33**, 175012 (2016), 1512.02864.

[97] H. Yu, R. X. Adhikari, R. Magee, S. Sachdev, and Y. Chen, Phys. Rev. D **104**, 062004 (2021), 2104.09438.

[98] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic, *Advances in quantum machine learning* (2015).

[99] S. Sharma, S. Sharma, and A. Athaiya, International Journal of Engineering Applied Sciences and Technology **04**, 310 (2020).

[100] D. Hendrycks and K. Gimpel, *Gaussian error linear units (gelus)* (2016), URL https://arxiv.org/abs/1606.08415.

[101] K. Fukushima, IEEE Transactions on Systems Science and Cybernetics **5**, 322 (1969).

[102] S. Ruder, arXiv e-prints arXiv:1609.04747 (2016), 1609.04747.

[103] J. Adejumo, *Gradient descent from scratch- batch gradient descent, stochastic gradient descent, and mini-batch gradient descent.*, https://medium.com/@jaleeladejumo/gradient-descent-from-scratch-batch-gradient-descent-stochastic-gradient-descent-ar accessed: 29-05-2024.

[104] D. P. Kingma and J. Ba, arXiv e-prints arXiv:1412.6980 (2014), 1412.6980.

[105] D. Jimenez Rezende and S. Mohamed, arXiv e-prints arXiv:1505.05770 (2015), 1505.05770.

[106] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference* (2021), 1912.02762.

[107] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv e-prints arXiv:1605.08803 (2016), 1605.08803.

[108] S. Ramasinghe, K. Fernando, S. Khan, and N. Barnes, arXiv e-prints arXiv:2102.03509 (2021), 2102.03509.

[109] R. T. Farouki and T. N. T. Goodman, Mathematics of Computation **65**, 1553 (1996), ISSN 00255718, 10886842, URL http://www.jstor.org/stable/2153723.

[110] R. Farouki and V. Rajan, Computer Aided Geometric Design **4**, 191 (1987), ISSN 0167-8396, URL https://www.sciencedirect.com/science/article/pii/0167839687900124.

[111] V. Klema and A. Laub, IEEE Transactions on Automatic Control **25**, 164 (1980).

[112] K. He, X. Zhang, S. Ren, and J. Sun, CoRR **abs/1512.03385** (2015), 1512.03385, URL http://arxiv.org/abs/1512.03385.

[113] K. He and J. Sun, arXiv e-prints arXiv:1412.1710 (2014), 1412.1710.

[114] R. K. Srivastava, K. Greff, and J. Schmidhuber, arXiv e-prints arXiv:1505.00387 (2015), 1505.00387.

[115] J. Lei Ba, J. R. Kiros, and G. E. Hinton, arXiv e-prints arXiv:1607.06450 (2016), 1607.06450.

[116] C. Gourieroux and A. Monfort, Journal of Econometrics **59**, 5 (1993), ISSN 0304-4076, URL https://www.sciencedirect.com/science/article/pii/0304407693900376.

[117] E. Cuoco et al., Mach. Learn. Sci. Tech. **2**, 011002 (2021), 2005.03745.

[118] V. Benedetto, F. Gissi, G. Ciaparrone, and L. Troiano, Appl. Sciences **13**, 9886 (2023).

[119] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, et al., *JAX: composable transformations of Python+NumPy programs* (2018), URL `http://github.com/google/jax`.

[120] P. Kidger and C. Garcia, arXiv e-prints arXiv:2111.00254 (2021), `2111.00254`.

[121] D. Ward, *Flowjax: Distributions and normalizing flows in jax* ([2024]), URL `https://github.com/danielward27/flowjax`.

[122] A. Zonca, L. Singer, D. Lenz, M. Reinecke, C. Rosset, E. Hivon, and K. Gorski, Journal of Open Source Software **4**, 1298 (2019), URL `https://doi.org/10.21105/joss.01298`.

[123] S. Antier, S. Agayeva, M. Almualla, S. Awiphan, A. Baransky, K. Barynova, S. Beradze, M. Blažek, M. Boër, O. Burkhonov, et al., Monthly Notices of the Royal Astronomical Society **497**, 5518 (2020), ISSN 0035-8711, `https://academic.oup.com/mnras/article-pdf/497/4/5518/33706665/staa1846_appendix_file.pdf`, URL `https://doi.org/10.1093/mnras/staa1846`.

[124] D. J. Thompson and C. A. Wilson-Hodge (2022), `2210.12875`.

[125] B. P. Abbott et al. (LIGO Scientific, Virgo), Astrophys. J. Lett. **892**, L3 (2020), `2001.01761`.

[126] I. Mandel and R. J. E. Smith, Astrophys. J. Lett. **922**, L14 (2021), `2109.14759`.

[127] B. P. Abbott et al. (LIGO Scientific, Virgo), Phys. Rev. Lett. **119**, 141101 (2017), `1709.09660`.

[128] L. Mazwi, S. Razzaque, and L. Nyadzani, Mon. Not. Roy. Astron. Soc. **531**, 2162 (2024), `2405.11650`.

[129] S. Morisaki, Phys. Rev. D **104**, 044062 (2021), `2104.07813`.

[130] B. Zackay, L. Dai, and T. Venumadhav (2018), `1806.08792`.

[131] S. Rosat and J. Majstorović, Phys. Rev. D **103**, 104052 (2021), URL `https://link.aps.org/doi/10.1103/PhysRevD.103.104052`.

[132] T. Dooney, L. Curier, D. Tan, M. Lopez, C. Van Den Broeck, and S. Bromuri (2024), `2401.16356`.

[133] R. Kumari and S. Srivastava, International Journal of Computer Applications **160**, 11 (2017).

A Additional results

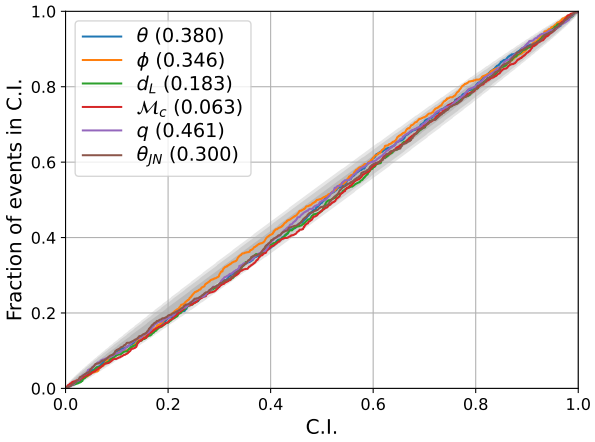


Figure 24: PP plots for the 70Hz network. The numbers behind the parameters are the p-values of that number. The grey area around the diagonal is the 99% confidence interval. The lines closely follow the diagonal, suggesting that the networks are unbiased.

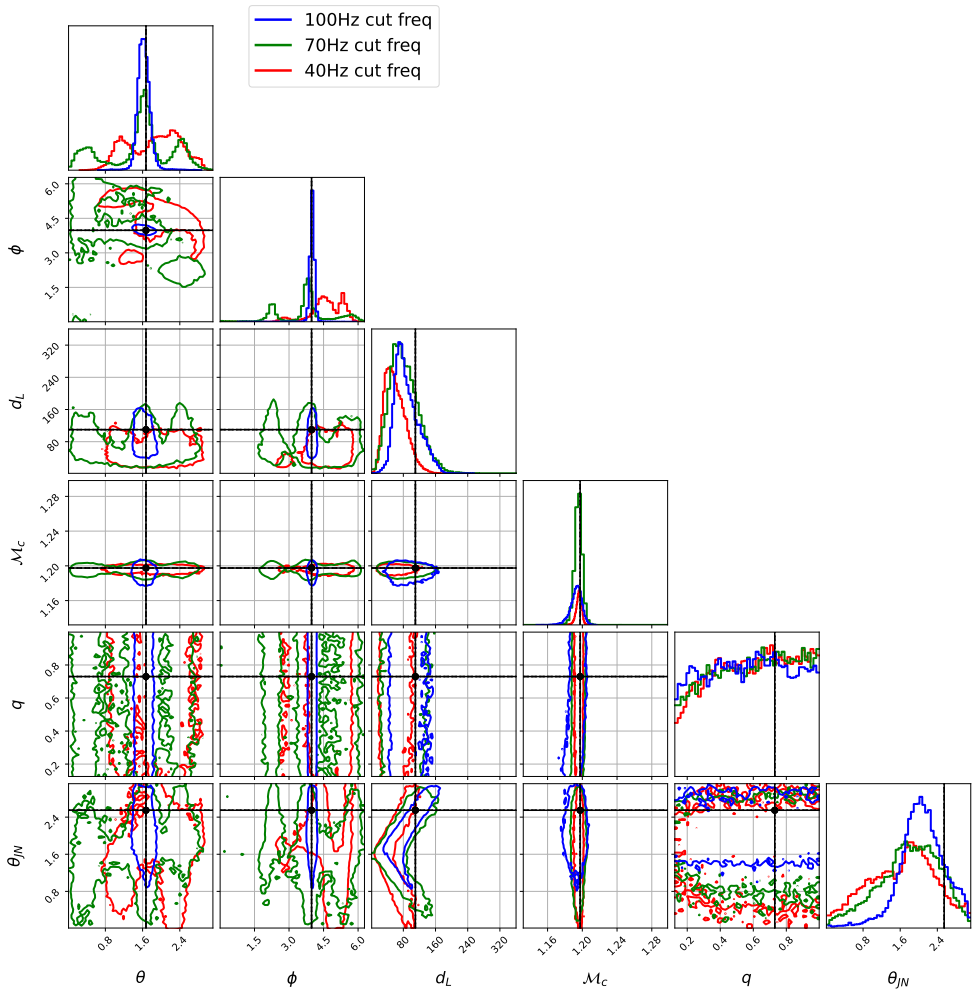


Figure 25: Corner plot of the estimated posteriors of the GW170817-like event. The PISNRs are 15.0, 19.2 and 22.5 for the 40, 70 and 100Hz networks respectively. The contours contain the 90% confidence intervals. The parameters are recovered reasonably well by all the networks.

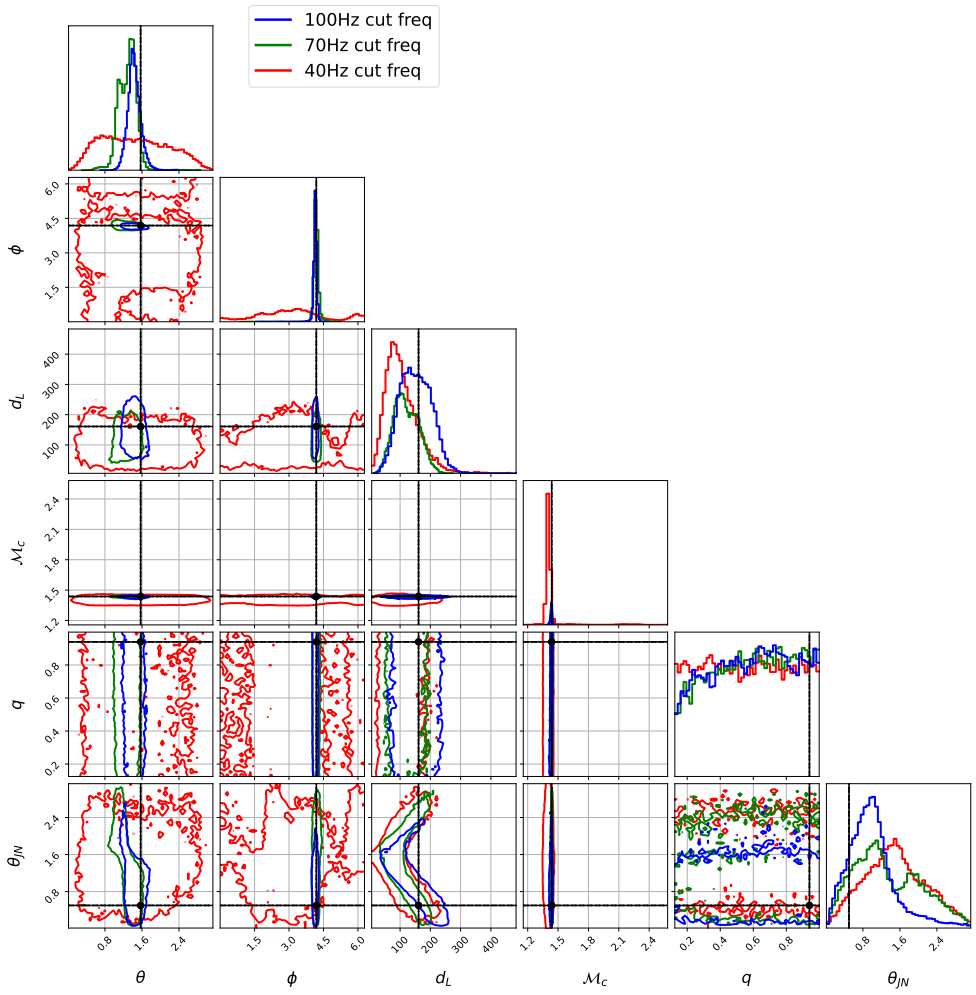


Figure 26: Corner plot of the estimated posteriors of the GW190425-like event. The PISNRs are 10.0, 16.7 and 19.6 for the 40, 70 and 100Hz maximum frequencies respectively. The contours contain the 90% confidence intervals. The chirp mass estimations are alright for all the networks, and the 100 Hz network is also able to find the inclination.

Acknowledgements

I want to thank Justin Janquart and Alex Kolmus for their excellent guidance and discussions during this project. You have both helped me tremendously in growing as a researcher and helped me peak my interest in pursuing a career in science. I would also like to thank Chris Van Den Broek for his supervision and guidance during the project. I also want to thank Jurriaan Langendorff for our excellent cooperation and countless pingpong matches during the project, keeping both our heads in the game. Lastly, I would like to thank Ruth Sanders for her continuous support and help!