Machine Learning Gravitational Waveforms for Binary Neutron Star mergers

Jacopo Tissino

2021-08-20

Abstract

This is the abstract.

Conventions and notation

$c = 1$ for sure.

Tensors with mixed indices should be written as $A^{\mu}{}_{\nu}$, but if $A_{\mu\nu}$ is symmetric then $A^{\mu}{}_{\nu} = A_{\nu}{}^{\mu}$.

Contents

1 Gravitational wave theory

1.1 Linearized gravity

The simplest way to discuss gravitational radiation is to consider linearized gravity on a flat Minkowskian background.

We give a brief overview, roughly following the path taken by Maggiore [Mag07, chapter 1].

This means that we assume that our spacetime admits a reference frame for which the metric is in the form

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \,, \tag{1.1}$$

where the value of the components $h_{\mu\nu}$ is small enough that we can work to first order in them. Any equation in this section includes an implicit "$+\mathcal{O}\left(h^2\right)$". We work in this *global inertial frame*.

In order to study the evolution of the perturbation $h_{\mu\nu}$ we need to solve the Einstein Field Equations for it to linear order. In a vacuum, they can be written as

$$\begin{aligned} G_{\mu\nu} &= R_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu} R = 0 &(1.2)\\ R_{\mu\nu} &= g^{\alpha\beta} R_{\alpha\mu\beta\nu} &(1.3)\\ R &= g^{\mu\nu} R_{\mu\nu} \,, &(1.4) \end{aligned}$$

where $R_{\alpha\mu\beta\nu} \sim \partial\Gamma + \Gamma\Gamma$ is the Riemann tensor, which is written in terms of derivatives and squares of Christoffel symbols Γ:

$$\Gamma^{\rho}_{\mu\nu} = \frac{1}{2} g^{\rho\lambda} \left(\partial_\mu g_{\nu\lambda} + \partial_\nu g_{\lambda\mu} - \partial_\lambda g_{\mu\nu} \right) \tag{1.5}$$

$$= \frac{1}{2}\eta^{\rho\lambda}\left(\partial_\mu h_{\nu\lambda} + \partial_\nu h_{\lambda\mu} - \partial_\lambda h_{\mu\nu}\right). \tag{1.6}$$

We use the fact that $\partial\eta = 0$, and keep only linear order terms.

Since the Christoffel symbols are of first order in the perturbation, the term $\Gamma\Gamma$ in the Riemann tensor is of second order and can be neglected. Therefore, the relevant components are

$$R_{\mu\nu\rho\sigma} = 2\eta_{\mu\lambda}\Gamma^{\lambda}_{\nu[\sigma,\rho]} = h_{\mu[\sigma,\rho]\nu} - h_{\nu[\sigma,\rho]\mu}\,, \tag{1.7}$$

which gives us the following expression for the Ricci tensor:

$$R_{\mu\nu} = \frac{1}{2}\left(h^{\sigma}{}_{\mu,\sigma\nu} + h^{\sigma}{}_{\nu,\sigma\mu} - h_{,\mu\nu} - \Box h_{\mu\nu}\right), \tag{1.8}$$

where $h = \eta^{\mu\nu}h_{\mu\nu}$ is the trace of the perturbation (computed with respect to the flat metric), while $\Box = \eta^{\mu\nu}\partial_\mu\partial_\nu$ is the flat space d'Alambertian.

This, in turn, allows us to write out the Einstein tensor:

$$G_{\mu\nu} = \frac{1}{2}\left(h^{\sigma}{}_{\mu,\sigma\nu} + h^{\sigma}{}_{\nu,\sigma\mu} - h_{,\mu\nu} - \Box h_{\mu\nu} - \eta_{\mu\nu}h^{\rho\lambda}{}_{,\rho\lambda} + \eta_{\mu\nu}\Box h\right). \tag{1.9}$$

This can be greatly simplified with two steps: first, we change variable from the perturbation $h_{\mu\nu}$ to the *trace-reversed* perturbation $\overline{h}_{\mu\nu} = h_{\mu\nu} - \eta_{\mu\nu}h/2$ — the name comes from the fact that $\eta^{\mu\nu}\overline{h}_{\mu\nu} = -h$.

This substitution allows us to write the Einstein tensor as

$$G_{\mu\nu} = -\frac{1}{2}\Box\overline{h}_{\mu\nu} + \overline{h}_{\alpha(\mu,\nu)}{}^{\alpha} - \frac{1}{2}\eta_{\mu\nu}\overline{h}_{\alpha\beta}{}^{,\alpha\beta}\,. \tag{1.10}$$

We will shortly show that it is possible, as a *gauge choice*, to set the divergence of the trace-reversed perturbation to zero: $\partial^\mu\overline{h}_{\mu\nu} = 0$ (see section 1.1.2). The gauge imposed by this choice is called the *Hilbert Gauge*,[1] which in terms of the regular perturbation reads

$$\partial^\mu h_{\mu\nu} - \frac{1}{2}\partial_\nu h = 0\,. \tag{1.11}$$

With this choice the Einstein tensor becomes simply

$$G_{\mu\nu} = -\frac{1}{2}\Box\overline{h}_{\mu\nu}\,, \tag{1.12}$$

so the general form of the Einstein equations to linear order will be

$$\boxed{\Box\overline{h}_{\mu\nu} = -16\pi G\, T_{\mu\nu}\Big|_{\text{linear}}\,, \qquad (1.13)}$$

where the stress-energy tensor is computed up to first order in the metric perturbation.

[1] Despite the name, this choice was first suggested by De Sitter to Einstein [Ken07], who had been previously trying to impose the gauge $|g| = 1$ [Ein16, page 688].

1.1.1 Transformations of the perturbation

The theory of General Relativity is constructed to be invariant under smooth changes of coordinates: under a map in the form $x \to x' = x'(x)$ (where $x'(x)$ is a diffeomorphism[2]).

Under such a coordinate transformation the metric transforms like any $(0,2)$ tensor:

$$g'_{\alpha\beta}(x') = \frac{\partial x^\mu}{\partial x'^\alpha} \frac{\partial x^\nu}{\partial x'^\beta} g_{\mu\nu}(x) \,. \tag{1.14}$$

A general transformation of this kind may break the condition that $g = \eta + h$ where h is small, so in order to preserve our framework we restrict ourselves to a small class of transformations.

One possibility is to consider infinitesimal transformations in the form

$$x^\mu \to x'^\mu = x^\mu + \xi^\mu(x) \,, \tag{1.15}$$

where ξ^μ is a vector field such that $\left|\partial_\mu \xi_\nu\right|$ is small — specifically, the condition to impose is that the first order in $\partial_\mu \xi_\nu$ should match the first order in $h_{\mu\nu}$.

This condition is all we need in order to write the transformation law for the perturbation: the full equation reads

$$\eta'_{\mu\nu} + h'_{\mu\nu} \approx \left(\delta^\alpha_\mu - \partial^\alpha \xi_\mu\right)\left(\delta^\beta_\nu - \partial^\beta \xi_\nu\right)\left(\eta_{\alpha\beta} + h_{\mu\nu}\right) , \tag{1.16}$$

Used $1/(1+x) = 1 - x + \mathcal{O}\left(x^2\right)$.

so the zeroth order contribution is $\eta'_{\mu\nu} = \eta_{\mu\nu}$, while the first order one is

$$\boxed{h'_{\mu\nu} = h_{\mu\nu} - 2\partial_{(\mu}\xi_{\nu)} \,,} \tag{1.17}$$

which is our transformation law for the metric perturbation.

We will also need a transformation law for the trace-reversed perturbation $\overline{h}_{\mu\nu}$: the trace transforms as $h' \to h - 2\partial_\mu \xi^\mu$, therefore the required law is

$$\overline{h}'_{\mu\nu} = \overline{h}_{\mu\nu} - 2\partial_{(\mu}\xi_{\nu)} + \eta_{\mu\nu}\partial_\alpha \xi^\alpha \,, \tag{1.18}$$

A second class of transformations is a subset of Lorentz boosts and rotations: substituting $\partial x^\mu / \partial x'^\nu = \Lambda_\nu{}^\mu$ into the transformation law (1.14) we find that the flat metric is unchanged, while

$$h'_{\mu\nu}(x') = \Lambda_\mu{}^\alpha \Lambda_\nu{}^\beta h_{\alpha\beta} \,, \tag{1.19}$$

which may remain in the class of small metric perturbations: this is not guaranteed, but it is true for a certain subset of boosts and for all rotations [Mag07].

Finally, the perturbation is invariant under shifts in the form $x'^\mu = x^\mu + a^\mu$.

[2] In the physics parlance this property is known as "diffeomorphism invariance", while a mathematician would call the kinds of transformations considered "isometries", since we ask that they preserve the metric structure of the manifold.

1.1.2 Gauge fixing

Now that we know how the perturbation $h_{\mu\nu}$ transforms under an infinitesimal transformation, we can use this to impose the condition we want — specifically, the Hilbert gauge (1.11).

The way to show that this is possible is to write out the way $\partial^\mu \overline{h}_{\mu\nu}$ transforms for an arbitrary choice of ξ, and to see that with an appropriate choice of ξ we can always map it to zero. The transformation reads

$$\begin{aligned}
\partial^\mu \overline{h}'_{\mu\nu} &= \partial^\mu \left(h_{\mu\nu} - 2\partial_{(\mu}\xi_{\nu)} \right) - \frac{1}{2}\partial_\nu \left(\eta^{\alpha\beta} \left(h_{\alpha\beta} - 2\partial_{(\alpha}\xi_{\beta)} \right) \right) && (1.20) \\
&= \partial^\mu \overline{h}_{\mu\nu} - \Box\xi_\nu - \partial_\nu \left(\partial^\mu \xi_\mu \right) + \partial_\nu \left(\partial^\mu \xi_\mu \right) && (1.21) \\
&= \partial^\mu \overline{h}_{\mu\nu} - \Box\xi_\nu \,. && (1.22)
\end{aligned}$$

Therefore, from any starting gauge we must only find a ξ_ν such that $\partial^\mu \overline{h}_{\mu\nu} = \Box\xi_\nu$, and we will be in the correct gauge. This can always be done, since the D'Alambert equation $\Box f = g$ can always be solved for f — if we needed to compute ξ_ν explicitly (which we typically do not) we could use the Green's function $G(z)$ for the operator, defined by $\Box G(z) = \delta^{(4)}(z)$.

While the equation is solvable, the solution is not unique: if we were to define an "inverse" of the D'Alambertian it would not be a function but a one-to-many relation. Specifically, while keeping fixed the value of $\Box\xi_\nu$ we can add any function ζ_ν to ξ_ν as long as $\Box\zeta_\nu = 0$. A trivial example is $\zeta_\nu = \text{const}$, but other wave-like choices are of more interest.

These still induce a transformation on $h_{\mu\nu}$ according to the usual law (1.17), and they can be used to further specify the form of the gravitational radiation.

In terms of **degrees of freedom**, the full perturbation $h_{\mu\nu}$ starts with 10 as any symmetric 4D, rank-2 tensor; the four Hilbert gauge conditions (1.11) reduce them to 6, while the four residual gauge conditions will allow us to reduce them to 2.

Doing so in full generality is not useful for us, let us instead fix the residual gauge in the specific context of a plane wave solution to the wave equation $\Box\overline{h}_{\mu\nu} = 0$.

1.1.3 Plane gravitational waves

From an analogy to electromagnetic theory it seems reasonable to work with an ansatz in the form

$$\overline{h}_{\mu\nu} = A_{\mu\nu} e^{ik_\alpha x^\alpha} \,, \tag{1.23}$$

where $A_{\mu\nu}$ is a constant symmetric tensor.

Imposing the wave equation sets $k_\alpha k^\alpha = 0$, and the Hilbert gauge (1.11) condition can be written as $A_{\mu\nu} k^\mu = 0$, where $A = \eta^{\mu\nu} A_{\mu\nu}$.

In this framework we can impose the residual gauge condition explicitly: a function which satisfies $\Box\zeta_\mu = 0$ is $\zeta_\mu = B_\mu \exp(i d_\alpha x^\alpha)$, where d_α is a null vector ($d_\alpha d^\alpha = 0$) while B_μ is a generic constant vector.

In these terms, the transformation equation (1.18) reads

$$\bar{h}_{\mu\nu} \to \bar{h}_{\mu\nu} + \left(-2iB_{(\mu}d_{\nu)} + i\eta_{\mu\nu}B_\beta d^\beta\right)e^{id_\alpha x^\alpha} \tag{1.24}$$

$$A_{\mu\nu}e^{ik_\alpha x^\alpha} \to A_{\mu\nu}e^{ik_\alpha x^\alpha} + \left(-2iB_{(\mu}d_{\nu)} + i\eta_{\mu\nu}B_\beta d^\beta\right)e^{id_\alpha x^\alpha}\,. \tag{1.25}$$

This tells us that if we set the vector d_α to be equal to k_α the amplitude $A_{\mu\nu}$ will transform according to the algebraic system

$$A_{\mu\nu} \to A_{\mu\nu} - 2iB_{(\mu}k_{\nu)}\,, \tag{1.26}$$

which allows us to impose four conditions on $A_{\mu\nu}$, one for each of the free components of B_μ. It is customary to choose $A = 0 = A_{0i}$: these are known together as the transverse-traceless (TT) gauge.

The condition $A = 0$ also means that $h = 0 = \bar{h}$: thus, form this point onward we can stop distinguishing between $h_{\mu\nu}$ and $\bar{h}_{\mu\nu}$, and for simplicity's sake we write the former.

If we orient our axes such that $\vec{k} = k^i$ is along the $\hat{z}$ direction (which means $k^\mu = (k, 0, 0, k)^\top$) the conditions can be written as

1. Hilbert gauge + traceless: $A_{\mu 0} + A_{\mu 3} = 0$;

2. traceless: $A = -A_{00} + A_{11} + A_{22} + A_{33} = 0$;

3. transverse: $A_{0i} = 0$.

The Hilbert gauge combined with the transverse conditions show that $A_{00} = 0$ as well, followed by $A_{13} = 0$, $A_{23} = 0$ and $A_{33} = 0$.

Finally, the traceless condition imposes $A_{11} = -A_{22}$. These conditions tell us that the plane gravitational wave must have the form

$$\boxed{h_{\mu\nu}(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \exp(ik_\alpha x^\alpha)\,,} \tag{1.27}$$

where $h_+ = A_{11}$ and $h_\times = A_{12}$.

1.1.4 Effect on test masses

The TT gauge we defined has the rather peculiar characteristic of "moving with the wave", so that the position of any observer initially at rest (so, such that $u^\mu(\tau = 0) = (1, \vec{0})^\top$) is unchanged: the geodesic equation evaluated at $\tau = 0$ reads

$$\left.\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\tau^2}\right|_{\tau=0} + \left.\Gamma^\mu_{\nu\rho}u^\nu u^\rho\right|_{\tau=0} = 0 \tag{1.28}$$

$$\left.\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\tau^2}\right|_{\tau=0} = -\Gamma^\mu_{00} = 0\,, \tag{1.29}$$

since the Christoffel symbols computed Γ^μ_{00} with the TT gauge perturbation (1.27) all vanish: $\Gamma^\mu_{00} = \eta^{\mu\nu}\left(2g_{\nu 0,0} - g_{00,\nu}\right)/2 = 0$.

Does this mean that gravitational waves are merely an artefact, and have no effect on particles? No, since while in the TT gauge the *positions* of the points do not change, the *distance* among them does.

In order to understand this effect we can make use of the geodesic equation [Car19, section 3.10], which states that the acceleration experienced between two geodesics whose four-velocities are both approximately u^μ, separated by a vector ξ^μ,[3] is

$$a^\mu = R^\mu_{\nu\rho\sigma} u^\nu u^\rho \xi^\sigma\,. \tag{1.30}$$

Let us consider the same geodesics as before, whose four-velocity is uniformly $u^\mu \equiv \left(1, \vec{0}\right)^\top$: the acceleration will then be given by the matrix product $R^\mu_{00\sigma}\xi^\sigma$.

These components of the Riemann tensor read [Car19, eq. 7.106]:

$$R^\mu_{00\sigma} = \frac{1}{2}\ddot{h}^\mu_\sigma\,. \tag{1.31}$$

Therefore, the temporal component of the acceleration is $a^0 \propto \ddot{h}^0_\sigma = 0$, while the spatial components read

$$a^i = \frac{1}{2}\ddot{h}^i_j \xi^j\,. \tag{1.32}$$

These equations can be explicitly solved, and a^i can be interpreted as $\ddot{s}^i$ for some notion of "position" of the points; note however that this is *not* coordinate position, but instead it denotes the changing distance between the points, which are stationary in the TT-gauge coordinates.

The positions of points at a fixed position (x, y, z) change (up to a phase) according to the expression

$$x(t) = x(t=0)\left(1 + h_+ e^{i\omega t}\right) + y(t=0) h_\times e^{i\omega t} \tag{1.33}$$

$$y(t) = y(t=0)\left(1 - h_+ e^{i\omega t}\right) + x(t=0) h_\times e^{i\omega t}\,, \tag{1.34}$$

which is shown graphically in figure 1 for four configurations: only h_+ or $h_\times$ being nonzero, or only $h_R \propto h_+ + ih_\times$ or $h_L \propto h_+ - ih_\times$ being nonzero. The last two are called *circular polarizations*.

[3] There is a technical note to be made here: in GR the notion of a vector between two points in the manifold is meaningless, however we can get around this problem by considering a one-parameter family of geodesics, and identifying the separation vector between them to be the tangent vector associated to the parameter.

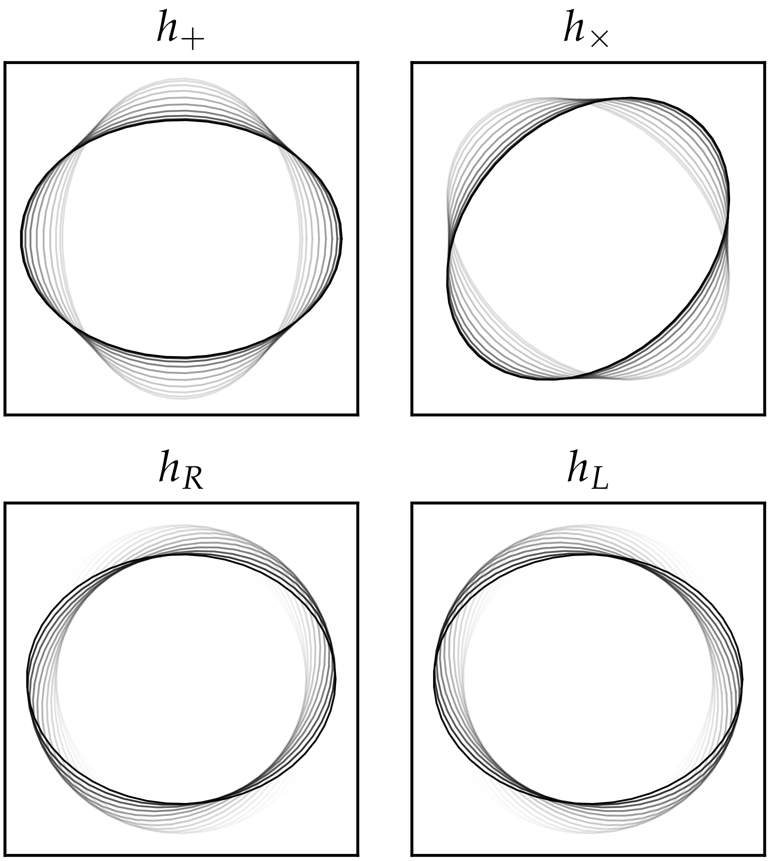

Figure 1: Polarizations of gravitational waves. Time evolution is represented though transparency: darker ellipses correspond to later points in time. The four cases are described in the text. We stress that in the TT gauge the ellipses do not represent changes in coordinate position but only in distance among points. In all cases the starting configuration is circular. The code generating this figure is available here.

1.1.5 The quadrupole formula

The lowest order contribution to the generation of gravitational waves can be calculated starting from the linearized Einstein equation (1.13). The Green's function method for the inversion of the D'Alambertian is a common technique: if we can find a function $G(z)$ such that $\Box G(z) = \delta^{(4)}(z)$, then we can do the following manipulation (denoting the gravitational constant as G_N for clarity):

$$\begin{aligned} \Box_x \overline{h}_{\mu\nu}(x) &= -16\pi G_N T_{\mu\nu}(x) && (1.35) \\ &= -16\pi G_N \int \mathrm{d}^4 y \, T_{\mu\nu}(y) \delta^{(4)}(x-y) && (1.36) \\ &= -16\pi G_N \int \mathrm{d}^4 y \, T_{\mu\nu}(y) \Box_x G(x-y) && (1.37) \end{aligned}$$

$$= \Box_x \left(-16\pi G_N \int \mathrm{d}^4 y \, T_{\mu\nu}(y) G(x-y) \right) \tag{1.38}$$

$$\bar{h}_{\mu\nu} = -16\pi G_N \int \mathrm{d}^4 y \, T_{\mu\nu}(y) G(x-y) \tag{1.39}$$

$$= 4 G_N \int \mathrm{d}^3 y \, T_{\mu\nu}(x^0 - |\vec{x} - \vec{y}|, \vec{y}) \frac{1}{|\vec{x} - \vec{y}|} \,, \tag{1.40}$$

Removing the D'Alambertian is valid insamuch as it will give us *a* solution for the wave equation, which will not be unique: $\Box f = \Box g$ does not imply $f = g$, but the reverse is true.

where in the last step we introduced the explicit expression for the Green's function of the D'Alambertian: $G(z) = -[z^0 > 0]/4\pi|\vec{z}|$[4] [Mag07, eq. 3.6].

We then have a formula for the trace-reversed perturbation, and it can be shown [Mag07, page 10] that far from the source we can recover the TT gauge perturbation by projecting this equation thanks to a tensor $\Lambda_{ij,kl}(\hat{n})$:

$$h_{ij}^{TT}(t, \vec{x}) = 4 G_N \Lambda_{ij,kl}(\hat{n}) \int \frac{\mathrm{d}^3 y}{|\vec{x} - \vec{y}|} T_{kl} \Big(t - |\vec{x} - \vec{y}|, \vec{y} \Big) \,, \tag{1.41}$$

where the tensor Λ is constructed to be a projector which sends 2-tensors into the subspace of traceless tensors, which are transverse to the direction defined by $\hat{n} = \vec{x}/|\vec{x}|$:

$$\Lambda_{ij,kl} = P_{ik} P_{jl} - \frac{1}{2} P_{ij} P_{kl} \qquad \text{where} \qquad P_{ij} = \delta_{ij} - n_i n_j \,. \tag{1.42}$$

This expression can be further simplified by making use of the assumption that the source is far away: the integral over $\mathrm{d}^3 y$ ranges over a region which has a maximum size of R, the scale of the source, while the wave is observed at a distance $r = |\vec{x}| \gg R$. This allows us to expand and neglect terms of the order R^2/r^2 and over;[5] the result we get is

$$h_{ij}^{TT} = \frac{4 G_N}{r} \Lambda_{ij,kl}(\hat{n}) \int \mathrm{d}^3 y \, T_{kl} \big(t - r + \vec{y} \cdot \hat{n}, \vec{y} \big) \,. \tag{1.44}$$

This equation uses the stress-energy tensor as computed without considering the higher-order effects of gravity on matter (this is the meaning of "linear" in the linear wave equation (1.13)); thus it is only a good approximation in the case of objects whose typical scale R is much larger, than, say, their Schwarzschild radius $2GM$. The problem with this assumption is that it fails to hold precisely for the sources which we are most interested in since, as we shall see, they give out some of the most easily detectable gravitational radiation: binary compact objects near coalescence. Let us forget about this issue for now.

For a gravitationally bound source like a binary, with total mass M and reduced mass μ, moving with speed v, the virial theorem dictates that

$$\frac{1}{2} \mu v^2 = \frac{1}{2} G \frac{\mu M}{r} \implies v^2 = \frac{GM}{r} \sim \frac{R_s}{r} \,. \tag{1.45}$$

[4] The square bracket here is the Iverson bracket [Knu92], which maps a boolean expression to 1 (if true) or 0 (if false); in this context it is thus equivalent to the Heaviside Theta: $[z^0 > 0] = \Theta(z^0)$.

[5] Specifically, we expand

$$|\vec{x} - \vec{y}| = \sqrt{x^2 + y^2 + 2x \cdot y} = r \sqrt{1 - 2 \frac{\hat{n} \cdot \vec{y}}{r} + \frac{y^2}{r^2}} \approx r \left(1 - 2 \frac{\hat{n} \cdot \vec{y}}{r} + \mathcal{O}\Big(R^2/r^2 \Big) \right) . \tag{1.43}$$

This means that the source not being very compact ($R_s \ll r$) is equivalent to it moving slowly ($v \ll 1$). Neither of these assumptions will hold in the end, but since we are using one we might as well use both.

In Fourier space, this amounts to saying that the typical frequencies ω for which the amplitude of the Fourier transform of the stress-energy tensor is large will satisfy $\omega \vec{y} \cdot \hat{n} \lesssim \omega R \ll 1$. This means that we can expand the exponential in the Fourier transform; in the time domain this amounts to expanding in time around the retarded time $t - r$:

$$T_{kl}(t - r + \vec{y} \cdot \hat{n}, \vec{y}) \approx T_{kl}(t - r, \vec{y}) + \vec{y} \cdot \hat{n} \partial_t T_{kl} + \frac{1}{2} (\vec{y} \cdot \hat{n})^2 \partial^2_{tt} T_{kl} + \mathcal{O}\Big((\vec{y} \cdot \hat{n})^3 \Big) . \tag{1.46}$$

This allows us to write the resulting wave as

$$h^{TT}_{ij}(t, \vec{x}) = \frac{4G_N}{r} \Lambda_{ij,kl} \left(S_{kl} + \sum_{L=1}^{\infty} \frac{1}{L!} \partial^L_t S_{kl|i_1 \dots i_L} n_{i_1} \dots n_{i_L} \right) , \tag{1.47}$$

To be computed at redarded time.

where the moments of the stress tensor are defined as

$$S_{kl|i_1 \dots i_L} = \int \mathrm{d}^3 y \, T_{kl} y_{i_1} \dots y_{i_L} . \tag{1.48}$$

We can also analogously calculate moments for the energy density T_{00}, which we denote $M_{i_1 \dots i_L}$, and for the momentum density T_{0i}, which we denote $P_{k|i_1 \dots i_L}$.

With integration by parts combined with the conservation of the (flat space!) stress-energy tensor $\partial_\mu T^{\mu\nu} = 0$ we can relate the M, P and S with equations such as [Mag07, eqs. 3.45–51]

$$\dot{M} = 0 \tag{1.49}$$

$$\dot{P}_i = 0 \tag{1.50}$$

$$\dot{M}_{ij} = 2P_{(i|j)} \tag{1.51}$$

$$S_{ij} = P_{i|j} . \tag{1.52}$$

These all tell us something interesting: $\dot{M} = 0$ and $\dot{P}_i = 0$ are energy and momentum conservation, which seem to tell us that there is no energy nor momentum loss from gravitational wave (GW) emission. This is an artifact, due to the assumptions of linear theory which neglect back-action on the source; fortunately we will still be able to compute the energy loss of the system.

The second useful fact is $S_{ij} = \ddot{M}_{ij}/2$: this allows us to write the lowest-order approximation of the expression of the wave from the source in terms of moments (1.47) as

$$h^{TT}_{ij}(t, \vec{x}) = \frac{2G_N}{r} \Lambda_{ij,kl} \ddot{M}_{kl} , \tag{1.53}$$

where

$$M_{kl}(t - r) = \int \mathrm{d}^3 y \, T_{00}(t - r, \vec{y}) y_k y_l . \tag{1.54}$$

since this expression only depends on the trace-free part of the moment M we can write it as a function of the traceless **quadrupole moment**

$$Q_{kl} = M_{kl} - \delta_{kl} M_{nn}/3 = \int \mathrm{d}^3 y \, \rho(t, \vec{y}) \left(y^i y^j - \frac{1}{2} \delta^{ij} y^2 \right) . \tag{1.55}$$

This gives rise to the quadrupole formula

$$\boxed{h_{ij}^{TT}(t, \vec{x}) = \frac{2G_N}{r} \Lambda_{ij,kl} \ddot{Q}_{kl}(t - r) . \tag{1.56}}$$

1.1.6 Energy loss through gravitational radiation

The problem of quantifying the energy carried by gravitational radiation is thorny: first of all, there is no universally valid way to split the perturbation from the background in the general case; also, at each point we can always apply the equivalence principle to recover flat spacetime up to first order. It is impossible to construct a *local*, gauge invariant stress-energy tensor $T_{\mu\nu}^{GW}$ for GW radiation: the limit of the energy density contained in any volume will always vanish as that volume goes to zero.

However, we *can* define a tensor through an averaging procedure over many wavelengths and periods of the wave. There are different ways to do so, but a common one is the Landau-Lifshitz pseudotensor:

$$t_{\mu\nu} = -\frac{1}{8\pi G} \left\langle R_{\mu\nu}^{(2)} - \frac{1}{2} \overline{g}_{\mu\nu} R^{(2)} \right\rangle_{\text{mesoscopic}} \tag{1.57}$$

$$= \frac{1}{32\pi G} \left\langle \partial_\mu h_{\alpha\beta} \partial_\nu h^{\alpha\beta} \right\rangle , \tag{1.58}$$

where $R_{\mu\nu}^{(2)}$ and $R^{(2)}$ are the components of the Ricci tensor and scalar which are quadratic in the perturbation $h_{\mu\nu}$; $\overline{g}_{\mu\nu}$ is the background metric, and the averaging procedure is done on scales (wavelengths/periods) which are (much) larger than the typical wavelengths of the gravitational radiation considered, but (much) smaller than the typical wavelengths of the background [Mag07, sections 1.4.2, 1.4.3].

This pseudotensor can be used to describe the way in which, on large enough scales, the presence of GWs does indeed curve spacetime. Also, we can compute the energy flux passing through a surface a large distance from the source: if we use the quadrupole formula (1.56) for the gravitational perturbation, we can give an expression for the emitted power in terms of the third derivatives of the quadrupole as [Mag07, eq. 3.3.2]

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{G_N}{5} \left\langle \dddot{Q}_{ij} \dddot{Q}_{ij} \right\rangle . \tag{1.59}$$

Dimensionally, $Q \sim \int \mathrm{d}^3 y \, \rho r^2$ has units of kgm^2; so $\dddot{Q}$ has units of $\mathrm{kgm}^2/\mathrm{s}^3 = \mathrm{W}$.

This means that the prefactor, which we wrote using $c = 1$, must really be $G_N c^n/5$ with units of inverse power, which implies $n = -5$. Numerically, it is the inverse of $5c^5/G_N \approx$

$2 \times 10^{53}\,\mathrm{W}$. This means that in order for GW emission to be efficient we must have a large value for $\dddot{Q}$: let us estimate it in terms of the typical size of the system, R, of its typical velocity $v = \Omega R$ and of its mass M. Each time derivative will roughly correspond to multiplication by a factor Ω, so $\dddot{Q} \sim \Omega^3 M R^2 = v^3 M / R$.

The power can then be estimated as

$$\frac{\mathrm{d}E}{\mathrm{d}t} \sim \frac{G_N}{5c^5} v^6 \frac{M^2}{R^2} = \underbrace{\frac{1}{5}\frac{c^5}{G_N}}_{\sim 10^{52}\,\mathrm{W}} \left(\frac{v}{c}\right)^6 \left(\frac{G_N M}{c^2 R}\right)^2 , \tag{1.60}$$

which tells us that the most significant sources of GW will be relativistic and compact. This prediction has been validated in 2015 with the first detection of GW from a binary black hole (BBH) system [Abb+16], and again in 2017 with the first detection of GW from a binary neutron star (BNS) system [Abb+17].

1.2 Compact binaries

We focus our attention towards a pair of inspiraling (compact) objects, which we will initially model as point masses. The first thing we need to do is to write an expression the amplitudes in the two polarizations $h_{+,\times}$ of the waves generated by a generic mass distribution with a mass moment M_{ij} [Mag07, eqs. 3.67–68]:[6]

$$h_+(t, \hat{n}) = \frac{G_N}{r}\left(\ddot{M}'_{11} - \ddot{M}'_{22}\right) \tag{1.61}$$

$$h_+(t, \hat{n}) = \frac{G_N}{r}\left(2\ddot{M}'_{12}\right) , \tag{1.62}$$

where M'_{ij} are the components of the mass moment tensor in a frame whose z' axis is aligned to the observation direction.

This frame is not the natural one with which to describe a binary system: typically, we would want to align the z axis of the coordinates with the rotation axis. The rotation matrix between two systems depends on two angles, however one of these is more important than the other: aligning the z axes can be accomplished by a rotation of angle ι, while another rotation of angle ϕ is needed in order to align the x and y axes as well. However, since (as we will discuss in a short while) the system is rotating in quasi-circular orbits, the second rotation only amounts to a time shift, or equivalently a phase in the oscillatory functions. For simplicity, we will neglect this phase freedom here and only include a variable phase at the end of the computations.

If the position of the bodies in the center-of-mass frame is

$$\vec{x}(t) = R \begin{bmatrix} \cos(\Omega t) \\ \sin(\Omega t) \\ 0 \end{bmatrix} , \tag{1.63}$$

[6] We could also write these in terms of the traceless quadrupole moment Q_{ij}; we follow the convention set forward by Maggiore [Mag07].

then the mass moments read $M_{ij} = \mu x_i(t) x_j(t)$ (where $\mu = m_1 m_2/(m_1+m_2)$ is the reduced mass of the system), and going through the computation yields [Mag07, eq. 3.332]

$$h_+(t) = \underbrace{\frac{4G_N \mu \Omega^2 R^2}{r}}_{A} \left(\frac{1+\cos^2\iota}{2} \right) \cos(2\Omega t) \tag{1.64}$$

$$h_\times(t) = \underbrace{\frac{4G_N \mu \Omega^2 R^2}{r}}_{A} \cos\iota \sin(2\Omega t) \tag{1.65}$$

$$A = \frac{4}{r} \left(\frac{G\mathcal{M}_c}{c^2} \right)^{5/3} \left(\frac{\pi f_{\text{gw}}}{c} \right)^{2/3} . \tag{1.66}$$

Approximating the motion of the bodies as circular during an orbit is called the *adiabatic approximation*, since it assumes that no energy is lost in each orbit (as that would deform the circle). The variation of the orbital frequency Ω, which we shall compute shortly, must satisfy $\dot{\Omega} \ll \Omega^2$. It is a rather good approximation for the early stages of the inspiral, as we shall see.

The last expression we wrote for the amplitude of the emission reintroduces factors of c, is expressed in terms of the emission frequency $f_{\text{gw}} = \omega_{\text{gw}}/2\pi = \Omega/\pi$ and of the *chirp mass*

$$\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1+m_2)^{1/5}} = \nu^{3/5} M \qquad \text{where} \qquad \nu = \frac{\mu}{M} . \tag{1.67}$$

1.2.1 Energy evolution

Writing the emission in this way allows us to give an expression for the emitted power

$$P = \frac{32}{5} \frac{c^5}{G} \left(\frac{G\mathcal{M}_c \omega_{\text{gw}}}{2c^3} \right)^{10/3} . \tag{1.68}$$

This allows us to describe the evolution of the orbit in this quasi-adiabatic context: the total energy of the binary system, by the virial theorem combined with Kepler's third law $\Omega^2 R^3 = GM$, reads

$$E = -\frac{G m_1 m_2}{2R} = -\frac{G m_1 m_2}{2} \frac{\Omega^{2/3}}{G^{1/3} M^{1/3}} = -\left(\frac{G^2 \mathcal{M}_c^5 \omega_{\text{gw}}^2}{32} \right)^{1/3} . \tag{1.69}$$

With these two expressions at hand, we can impose $P = -\dot{E}$, which we can write in the form $\dot{\omega}_{\text{gw}} \propto \omega^n$ for some n: since $\dot{E} \propto \omega_{\text{gw}}^{-1/3} \dot{\omega}_{\text{gw}}$, n will be $11/3$, and the specific expression will look like

$$\dot{\omega}_{\text{gw}} = \frac{12}{5} \sqrt[3]{2} \left(\frac{G\mathcal{M}_c}{c^3} \right)^{5/3} \omega_{\text{gw}}^{11/3} . \tag{1.70}$$

This equation is in the form $\omega^{-11/3}\,\mathrm{d}\omega = K\,\mathrm{d}t$; integrating from point 0 to point 1 we find $3K(t_1 - t_0)/8 = \omega_0^{-8/3} - \omega_1^{-8/3}$; if we treat this as an initial value problem we can see that as t_1 increases there is only a finite "budget" on the right-hand side: when $3K(t_1 - t_0)/8 = \omega_0^{-8/3}$, $\omega_1^{-8/3}$ must go to 0 so ω_1 must diverge. The divergence itself is unphysical since, besides corresponding to a breakdown of some of our approximations, it comes about when considering point masses, while compact objects have a finite size and at some point collide.

Nevertheless, the moment of divergence happens close enough to the actual merger of the objects, therefore it is useful to call it t_c and to define a time coordinate as $\tau = t_c - t$. In terms of this, the solution to the equation reads

$$f_{\mathrm{gw}}(t) = \frac{\omega_{\mathrm{gw}}(t)}{2\pi} = \frac{1}{2\pi}\left(\frac{3K\tau}{8}\right)^{-3/8} = \frac{1}{\pi}\left(\frac{5}{256\tau}\right)^{3/8}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{-5/8}. \tag{1.71}$$

At this point we can notice an interesting scaling property: if we map $f_{\mathrm{gw}} \to \alpha f_{\mathrm{gw}}$, $\tau \to \tau/\alpha$ and $\mathcal{M}_c \to \mathcal{M}_c/\alpha$ (or, specifically, we divide each mass by α) the equation still holds! This result turns out to hold in general, not only at the linear, quadrupole level. [FIND CITATION FOR THIS STATEMENT]

This expression for f_{gw} can be substituted into the gravitational waveforms (1.64); we must also update the phase term, since $2\Omega t$ is the phase for a uniformly circular orbit: we will want to use integrate the angular velocity, to find

$$\Phi(t) = \int^t \omega_{\mathrm{gw}}(\widetilde{t})\,\mathrm{d}\widetilde{t} = 2\int^t \Omega(\widetilde{t})\,\mathrm{d}\widetilde{t} \tag{1.72}$$

$$= -2\left(\frac{5G\mathcal{M}_c}{c^3}\right)^{-5/8}\tau^{-5/8} + \text{const}\,. \tag{1.73}$$

In terms of this, the amplitude in the two polarizations will read [Mag07, eqs. 4.31–32]

$$h_+(t) = \frac{1}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/4}\left(\frac{5}{c\tau}\right)^{1/4}\left(\frac{1+\cos^2\iota}{2}\right)\cos\Phi(t) \tag{1.74}$$

$$h_\times(t) = \frac{1}{r}\left(\frac{G\mathcal{M}_c}{c^2}\right)^{5/4}\left(\frac{5}{c\tau}\right)^{1/4}\cos\iota\sin\Phi(t)\,. \tag{1.75}$$

As we will discuss, it is convenient to have expressions for the Fourier transforms of these. An analytic computation of the integrals is intractable, but we can make use of a technique known as stationary phase approximation (SPA), which is discussed in detail in appendix A. The final expressions for the Fourier-domain waveforms are equations (A.18).

1.2.2 Parameters for a CBC waveform

The waveform we wrote explicitly depends on:

1. the chirp mass $\mathcal{M}_c$, defined in (1.67);

2. the distance r, which in a more general cosmological context should be replaced by the *luminosity distance* D_L;

3. the inclination ι, which is the angle between the observation direction and the total angular momentum of the system.

These are only some of the parameters which must be considered when discussing a CBC. We will now discuss the full set of parameters which can be used to fully describe the binary system generating the waveform [BGB21, eq. 21].

1.2.3 Arrival time and initial phase

The time at which any given waveform arrives at Earth is arbitrary, as is the global phase of the waveform. In practice, one will typically analyze the output from a detector in batches, and in each of these the analysis will be performed in Fourier space. Therefore, it is relevant to see how the Fourier-domain waveform responds to a time and phase shift: what is the transform of $f(t-t_0)e^{i\varphi_0}$? It comes out to be

$$\int_{\mathbb{R}} f(t-t_0)e^{i\varphi_0}e^{i\omega t}\,\mathrm{d}t = e^{i\omega t_0 + i\varphi_0} \underbrace{\int_{\mathbb{R}} f(t-t_0)e^{i\omega(t-t_0)}\,\mathrm{d}(t-t_0)}_{\widetilde{f}(\omega)}\,, \tag{1.76}$$

so a time shift corresponds to the addition of a linear term to the phase, while the phase can be directly moved from the time to the frequency domain.

1.2.4 Sky position and polarization

The wave will be coming from a sky position, which we can describe with respect to a given coordinate system through two angles; it is convenient to use right ascension α and declination δ. Also, the polarization of the gravitational waves can have an arbitrary angle with respect to the observation direction: we denote this angle as ψ.

1.2.5 Masses and mass ratio

The waveform to the lowest order we described only depends on the chirp mass $\mathcal{M}_c$, but as we will see later at higher order there appears a dependence on the ratio of the two masses: $q = m_1/m_2$, where m_i are the masses of the two compact objects. It is customary to fix one of the two masses to always be the largest, therefore constraining $q \lesseqgtr 1$; we choose the convention $q \geq 1$.

A possible parametrization for the two masses is $(\mathcal{M}_c, q)$, another is (M, q): these are equivalent, but in the following work we will typically use the latter because of the aforementioned scaling property of the waveforms: we conventionally work with respect to the dimensionless frequency Mf and the dimensionless time t/M in order to have one less parameter.

1.2.6 Spin

Compact objects can spin, and the spin of each of the two is a vector with three different components. These vectors $\vec{S}_i$ have units of angular momentum, but they are typically rescaled as

$$\vec{\chi}_i = \frac{c\vec{S}_i}{Gm_i} \in [-1, 1] \text{ in magnitude}\,. \tag{1.77}$$

The constraint of $|\chi_i| < 1$ can be approached by realistic models of BHs, while realistic models of NSs typically are constrained to spin much less, at the most $|\chi_i| \lesssim 0.7$ [LL11]. This constraint is hard to experimentally verify since one is faced with the degeneracy between the components $\chi_{z,i}$ aligned with the direction of the angular momentum and the mass ratio q; in the analysis of the merger GW170807 two sets of prior distributions for the spins were used, $|\chi| < 0.05$ and $|\chi| < 0.89$, for this reason [Abb+17]. We have theoretical reasons to believe that the low-spins priors might correspond to a more physically meaningful scenario, but the degeneracy means that the high-spin case is not experimentally excluded, so we must still consider it a possibility.

1.2.7 Tidal polarizability of neutron stars

If our compact objects are not black holes, they might be able to deform. The astrophysically motivated scenario we think of is that of neutron stars, but the following characterization can also apply to any extended compact objects. For concreteness' sake, we will refer to NSs.

This is a complicated process, but we can try to capture its most significant part by limiting ourselves to the quadrupole order of the deformation of the star. The discussion of this effect is clearest in the Newtonian context, but it can be generalized to general relativity (GR) [Mag18, section 14.4.1].

Tidal effects are described by the tidal tensor $\mathcal{E}_{ij} = -U_{,ij}$, the traceless[7] Hessian of the Newtonian potential U. In the weak-field relativistic case, this corresponds to part of the Riemann tensor: $\mathcal{E}_{ij} = c^2 R_{0i0j}$.

We can describe the effects on the deformed star of such a tidal stress by looking at its quadrupole moment Q_{ij} (1.55). To linear order and neglecting any time dependence, we will have the relation

$$\mathcal{E}_{ij} = -\lambda Q_{ij}\,. \tag{1.78}$$

find better explanation for minus sign here and in def of tidal tensor

This parameter λ describes the deformability of the star; if it is smaller the star deforms less in response to tidal perturbations. In practice the parameter used is not λ but two ways to rescale it: the $l = 2$ (quadrupole) **Love number**

$$k_2 = \frac{3}{2}\frac{G\lambda}{R^5}\,, \tag{1.79}$$

[7] This tidal tensor describes the effect of one star's gravitational field on the other, so its source is not where we compute it, which is a convoluted way of saying that it should be taken to be a solution of $\nabla^2 U = 4\pi\rho$ in vacuum: therefore, $\nabla^2 U = -\operatorname{Tr}\left[\mathcal{E}_{ij}\right] = 0$.

where R is the radius of the deformed neutron star (NS), or the **tidal deformability**

$$\Lambda = \frac{2}{3} k_2 \left(\frac{Rc^2}{Gm} \right)^5 = G\lambda \left(\frac{c^2}{Gm} \right)^5 , \tag{1.80}$$

where m is the mass of the deformed NS.

The Love number is useful since it is an expansion parameter for the gravitational field of the deformed NS. This field has contributions from both its quadrupole moment, which is due to the tidal deformation from the other star, and from the gravitational field of the other star which we can expand in terms of the tidal field around the center of the deformed star: $U_{\text{ext}} \approx -\mathcal{E}_{ij} x_i x_j / 2$.

The

1.3 Data analysis

2 Machine Learning

2.1 Principal Component Analysis

After downsampling, a waveform used by `mlgw_bns` is described by several hundred points. It is convenient to reduce this number in order for the neural network to be faster. We are able to do so by making use of the fact that the components of the high-dimensional vector representing the waveform are correlated.

The technique of principal component analysis (PCA) is quite general,[8] so let us describe it in general terms, and then apply it to our specific problem.

2.1.1 General method

We start with a dataset of N points in $\mathbb{R}^D$, which we denote by $\{x^i\}_{i=0}^{N-1}$. We need D floating point numbers to represent each of these points.

If we can find a k-dimensional hyperplane in $\mathbb{R}^D$, with $k \ll D$, such that our points are never very far from this subspace, we can substitute the D-dimensional parametrization of the points for a k-dimensional one by approximating each point by its orthogonal projection onto the k-dimensional hyperplane. We will make a certain error in this process: specifically, if $P_k(x_i)$ denotes the projection of the point onto this hyperplane, the error (computed

[8] The utility and relative simplicity of this technique might make one think it quite old, while in fact it was developed only in 1901 by Karl Pearson [Pea01; MN17], ànd it started to see wider use once availability of computers became widespread.

according to the Euclidean distance among points[9]) can be quantified by

$$\text{error}(k) = \sum_{i=0}^{N-1} \left\| x_i - P_k(x_i) \right\|^2 . \tag{2.2}$$

The algorithm of PCA allows us to determine which hyperplane minimizes this error.

The first step is to center the data: we compute their mean $\overline{x}$, and work with the dataset $y_i = x_i - \overline{x}$. Because of this, we can say that the k-dimensional hyperspace is now a *subspace* with respect to y. Computationally, we keep the mean $\overline{x}$ saved and add it to the reconstructed data y.

Let us now consider the $k = 1$ case: we want to project the data onto a single line, which we can parametrize as the span of a unit vector w. Therefore, what we want to minimize is $\sum_i \left\| y_i - (y_i \cdot w) w \right\|^2 = \sum_i \left(\left\| y_i \right\|^2 - (y_i \cdot w)^2 \right)$, and we can do so by maximizing $\sum_i (y_i \cdot w)^2$.

Therefore, the best 1-dimensional subspace is the direction of maximum variance:

$$w = \operatorname*{argmax}_{w \in \mathrm{S}^{D-1}} \sum_i (y_i \cdot w)^2 . \tag{2.3}$$

Now comes the clever idea of PCA: we can reformulate this argmax problem as an eigenvalue problem for the covariance matrix of the data:

$$\mathsf{C} = \frac{1}{N} \sum_i y_i y_i^\top . \tag{2.4}$$

A unit eigenvector w of this matrix will satisfy $Cw = \lambda w$ for its eigenvalue λ; and we can recover the eigenvalue λ from this equation by computing $w^\top C w = \lambda w^\top w = \lambda$; making the covariance matrix explicit allows us to see that

$$\lambda = w^\top C w = \frac{1}{N} \sum_i (y_i \cdot w)^2 ; \tag{2.5}$$

which is precisely the quantity we wanted to maximize: therefore, the best one-dimensional subspace is precisely the largest eigenvector of the covariance matrix.

If we make the further observation that the covariance matrix is symmetric and positive definite, and can therefore be orthogonally diagonalized, we are almost done: we can generalize to arbitrary k moving one vector at a time. To find the second vector to span the subspace we can restrict ourselves to the subspace $w^\perp$ and apply the same procedure as

[9] One might object here: the Euclidean distance among points is hardly relevant for our practical application! Fortunately, as we will later discuss, the PCA reconstruction of the points is efficient according to the Wiener distance as well as according to the Euclidean one. This might be understood heuristically by thinking of the fact that, when looking at waveforms which are quite close in terms of both distances, the linear approximation

$$\text{Wiener distance}(a_i, b_j) \approx \sqrt{g_{ij} a_i b_j} \tag{2.1}$$

for some metric g_{ij}, where a_i and b_j denote the vectors representing the two waveforms. Now, we do not know the precise form of g_{ij} (and, of course, we could not give a unique expression since it varies with detector noise), but as long as the metric is not pathological convergence in the Euclidean distance will imply convergence for this alternative distance.

before, this tells us that the optimal two-dimensional subspace is the span of the first two eigenvectors of the covariance matrix, and so on.

In order to perform a reduction onto a k-dimensional subspace, then, we need to calculate the unit eigenvectors $\{w_i\}_{i=0}^{k-1}$ corresponding to the k largest eigenvalues; we can understand these as the columns of a $D \times k$ matrix V, which we can then use to construct the projection matrix onto the k-dimensional subspace.

In terms of the D-dimensional coordinates, the projection matrix is $VV^\top$: its application to a vector y can be written as

$$P_k(y) = VV^\top y = \left[\begin{array}{ccc} w_1 & \cdots & w_k \end{array} \right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_k \end{array} \right] y = \sum_{i=0}^{k-1} (w_i \cdot y) w_i \in \mathbb{R}^D \,. \tag{2.6}$$

For the purpose of dimensionality reduction, however, we are not interested in this projection, which still yields a D-dimensional vector: this vector lies in a k-dimensional subspace, therefore we express it with only k coordinates, by computing $V^\top y \in \mathbb{R}^k$.

If the

If we approach the problem by diagonalizing the covariance matrix C, the computational complexity in the worst case scenario is $\mathcal{O}\left(D^3\right)$, since it involves the diagonalization of a $D \times D$ matrix.

2.1.2 PCA for waveforms

After downsampling and computing the residuals from the post-Newtonian (PN) waveform, we are left with a waveform described by a vector of a few hundred points for the amplitude and similarly for the phase.

It would be possible to perform PCA separately for these two vectors; however if we combine them into a single one we can exploit any existing correlation between the amplitude and phase residuals. In the worst case scenario — amplitude and phase residuals being completely uncorrelated — this procedure will perform exactly like separating the PCA into two. So, we want to combine amplitude and phase residuals into a single vector.

A simple way to do so is to simply “append” one vector to the other, and therefore consider the waveform as a vector in $\mathbb{R}^{D_A + D_\varphi}$. There is an issue with this procedure: PCA optimizes Euclidean distance, so the scaling of the amplitude and phase residuals becomes relevant. Fortunately, this carries no physical meaning: because of the way we have defined the residuals [INSERT REF], it is equivalent to a change in the basis for the logarithm.

Heuristically, we want the typical scale of the phase residuals to roughly match that of the amplitude residuals. This can be simply achieved by multiplying either part of the vector by a constant, which can be tuned in order to optimize the reconstruction error. It was found that, after using natural logarithms for the amplitude residuals, dividing the phase residuals by roughly 200 yields the best results.

[INSERT FIGURE FOR PCA RECONSTRUCTION ACCURACY]

2.2 Neural Networks

After the reduction of dimensionality through PCA, we are left with the task of approximating the map between the parameters of the system generating the waveform, θ_i, and the k principal components.

In the work of Schmidt et al. [Sch+20] this was accomplished through a Mixture of Experts model, which amounts to a fit to a polynomial expression of the parameters; here instead we reconstruct the function with a neural network.

NNs are known to be able to approximate arbitrary functions [Nie15], and in practice they are quite versatile and usually not prone to overfitting. In the next section we will describe the architecture of a multi-layer perceptron regressor, the kind of network used by `mlgw_bns`.

2.2.1 Multi-layer perceptrons

This architecture is built in order to solve the problem of reconstructing a map from an input, in $x_i \in \mathbb{R}^n$, to an output in $y_i \in \mathbb{R}^k$.

We introduce a *hidden layer* between the input and the output. This consists of a certain number m of "neurons", which can be more or less activated as a function of the inputs. Specifically, the j-th neuron in the hidden layer will have an activation level given by the expression:

$$z_j = \sigma\left(w_{ji}^{(1)} x_i + b_j^{(1)}\right), \tag{2.7}$$

where $\sigma\colon \mathbb{R} \to \mathbb{R}$ is called the *activation function*, while the parameters $w_{ji}^{(1)}$ and $b_j^{(1)}$ are respectively called the *weights* and the *bias* (for the first hidden layer).

We shall discuss the reasons one might have for choosing different activation functions later; for now let us say that it is typically differentiable almost everywhere, and it achieves low values for low inputs and high values for high inputs. A common choice, for example, is a logistic sigmoid in the form $\sigma(z) = (1 + e^{-z})^{-1}$.

The weights and biases of the network are free parameters, real numbers which will need to be tuned by the training process.

Once the network has computed the activations z_j for our single hidden layer, it can compute the activations for the output: in this last stage we use no activation function, and the output of the network is simply

$$y_\ell = w_{\ell j}^{(2)} z_j + b_\ell^{(2)} = w_{\ell j}^{(2)} \sigma\left(w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b_\ell^{(2)}. \tag{2.8}$$

In the end, therefore, the number of free parameters of the network is $nm + m$ for the first layer and $mk + k$ for the second, so $m + k + m(n + k)$ in total.

Adding more layers is not conceptually different, it only amounts to applying the procedure described by equation 2.7 again to the result of the first layer, and so on; each time we use the activation function, except for the output layer.

The activation function is crucial for our network to be able to capture nonlinearities: if we were to remove it, with any number of layers the network would still be a linear function of the inputs.

The implementation we use for `mlgw_bns` is the one provided by `scikit-learn` [Ped+11].

2.2.2 Training

The network is able to reconstruct our function as long as the weights and biases are appropriately set: how do we train it to ensure this?

We need to assign a loss function to the output of the network. Here, simplicity trumps accuracy — the "true" error we might like to work with is given by the Wiener distance among waveforms reconstructed from their PCA components, but in order to efficiently train the network we need something easier to compute. The typical error chosen is quadratic in the Euclidean distance, since its analytical derivative is easy to compute:

$$\text{error}(y) \propto \sum_{\text{training data}} \left\| y_{\text{predicted}} - y_{\text{true}} \right\|^2 . \tag{2.9}$$

We also add an error term in the form $\alpha \|W\|^2$, where α is a non-negative hyperparameter (typically chosen to be small) and $\|W\|^2$ is the L2 norm of the weight tensor: this term is known as a "regularizer", it penalizes complex models. The parameter α is optimized in the hyperparameter training procedure, and its prior distribution is taken to be a log-uniform one between 10^{-7} and 10^{-5}.

We are then using the Euclidean distance among the ys (which for us will be PCA component vectors) as a measure of the performance of the network; this works well enough, but we can make an improvement by noticing that the first PCA components are responsible for more of the variance (and thus more of the distance) between data points. Therefore, we can improve the performance by having the network learn the distance among the rescaled

$$\text{PC}_i \lambda_i^{\kappa} , \tag{2.10}$$

where λ_i are the eigenvalues corresponding to the principal components, while $\kappa > 0$ is a hyperparameter. The prior distribution for κ is a log-uniform one between 10^{-3} and 1.

2.2.3 Backpropagation

Once we have our cost function, we need a rule to change the weights w_{ij}^l and biases b_j^l of our network[10] according to the variation of the cost function. We would like to implement some sort of gradient descent algorithm, updating weights and biases by

$$\Delta w_{ij}^l = -\eta \frac{\partial C}{\partial w_{ij}^l} \qquad \text{and} \qquad \Delta b_j^l = -\eta \frac{\partial C}{\partial b_j^l} , \tag{2.11}$$

where η is called the *learning rate*.

[10] We have added an index l for the layer: so, w_{ij}^l is the weight that the j-th neuron in the l-th layer gives to input i.

So, we need to compute the gradients $\partial C \Big/ \partial w^l_{ij}$ and $\partial C \Big/ \partial b^l_j$. The first idea one might have to do so is to approximate them to first order, doing something like

$$\frac{\partial C}{\partial w^l_{ij}} \approx \frac{C(w + \epsilon e^l_{ij}) - C(w)}{\epsilon}\,, \tag{2.12}$$

where w is the full weight tensor, while ϵe^l_{ij} represents a small increment to that particular weight, e being a "unit tensor".

This strategy turns out to be unfeasible because of its computational complexity: the computation of the cost requires a full pass-through of the network, requiring at least M floating point operations (additions or multiplications) where M is the number of free parameters. This would need to be done to compute the update of each of these parameters, so the number of operations needed to perform a single step of the gradient descent for the full network would be at least M^2.

The backpropagation algorithm is a clever idea which allows us to compute the gradient with only a forward pass through the network followed by a backward pass, exploiting the chain rule and the way the weight matrices affect each other iteratively.

The algorithm can be summarized by the following equations [Nie15, chapter 2]:

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \tag{2.13}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \tag{2.14}$$

$$\delta^l_j = \left(\sum_i w^{l+1}_{ij} \delta^{l+1}_i \right) \sigma'(z^l_j) \tag{2.15}$$

$$\delta^L_j = \frac{\partial C}{\partial a^L_j}\,, \tag{2.16}$$

where $a^l_j = w^l_{ji} z_i + b^l_j$ is the activation for the j-th neuron of layer l, while $z^l_j = \sigma(a^l_j)$ is the output of the activation function for a^l_j.

It is important to note that the backpropagation equations are written without the Einstein summation convention: "$x_j = y_j z_j$" means that the j-th component of the vector x is calculated by multiplying the j-th components of the vectors y and z. This means that we are not computing matrix products, but instead the element-wise Hadamard product.

Working backwards, the last equation tells us how to compute the error corresponding to the output layer, denoted with L. We can compute the derivative of the cost function with respect to the activation analytically if, for example, we are using a simple quadratic cost function as described in equation 2.9.[11]

The second-to-last equation tells us how to compute the error δ^l_j of a layer if we know the error of the following layer. It is an application of the chain rule. We compute the δ^l_j

[11] The expression in [Nie15] differs from this one since he applies the activation function to the last layer, which one should do for a classification algorithm but not for a regression algorithm.

iteratively going backwards through the network, and having done so we can recover the derivative of the cost with respect to the weights and biases by combining the error δ_j^l with the activations of each layer, a_k^{l-1}, as described by the first two equations.

2.2.4 Stochastic gradient descent

The method for gradient descent described by equation 2.11 is still slow if we try to compute the gradient of the cost function by using all the training data we have. It turns out to be more efficient to compute the gradient by only looking at a single example or a small batch of them, chosen randomly: this idea is known as *stochastic gradient descent*.

This typically allows for much faster convergence of the training process.

The algorithm used for the training of the network in `mlgw_bns` is Adam [KB17; Rud16], short for "Adaptive Moment Estimation", as implemented in `scikit-learn` [Ped+11].

The general idea of this algorithm is, first, to not move directly in the direction of the gradient, but instead to keep a running, exponentially weighted average of it, and move in *that* direction. An example where we can imagine this could be useful is if the cost function landscape exhibits a "canyon", with a low slope in a long and narrow central region and steep walls. A direct move in the direction of the gradient might mean we "bounce" between the walls a lot without being able to settle in the middle, while averaging allows the direction of our movement to be smoothed and perhaps fall in the middle region.

The second aspect is to make the step-size adaptive, as opposed to it being strictly proportional to the gradient. A "signal-to-noise ratio" is estimated through the square of the gradient, and it is used to scale the step based on how confident we might be in it being meaningful or not.

2.3 Hyperparameter optimization

Our network will depend on several hyperparameters, such as the number and size of the layers or the learning rate; we want to get as close as possible to the optimal choice of these for the reconstruction of the function mapping the binary system parameters to the principal components to be fast as well as accurate.

This optimization is accomplished through a multi-objective tree-structured Parzen estimator (MOTPE), as described in Ozaki et al. [Oza+20] and as implemented through the Optuna API [Aki+19].

Here we will summarize the mechanism through which a single objective tree-structured Parzen estimator (TPE) works, as originally described in Bergstra et al. [Ber+11, section 4], since the generalization to the multi-objective case [Oza+20] is rather mathematically involved but not conceptually dissimilar from the single-objective case.

We can abstract away the neural network as a function f which, after being given a set of hyperparameters $\vec{x}$, outputs a cost y, which we want to minimize. The evaluation of f is quite costly: it involves the training of the network and a full evaluation of its performance (in our case, the reconstruction of the waveforms and a computation of the Wiener distance to their true counterparts).

So, we want to find a value of $\vec{x}$ which minimizes y with as few evaluations of f as possible. The parameters $\vec{x}$ will be given certain prior distributions initially, from which their values will be drawn randomly. Let us then suppose we already have a set of observations $\{(\vec{x}_i, y_i)\}_i$, and we want to find the best possible new value of $\vec{x}$.

We choose a certain quantile γ, say 15%, and select a fraction γ of the best observations we have. This allows us to find a y^* such that $\mathbb{P}(y < y^*) = \gamma$.

Then, we approximate the probability density $p(\vec{x}|y)$ as follows:

$$p(\vec{x}|y) = \begin{cases} \ell(\vec{x}) & y < y^* \\ g(\vec{x}) & y \geq y^* \,. \end{cases} \tag{2.17}$$

We are condensing the y-dependence onto a binary choice between "good" observations, modelled by $\ell(\vec{x})$, and "bad" observations, modelled by $g(\vec{x})$. These two functions can then be estimated by making use of the observations we have in the $y < y^*$ and $y \geq y^*$ cases.

This, crucially, allows us to calculate the **expected improvement** associated with a certain parameter set $\vec{x}$:

$$\mathrm{EI}_{y^*}(\vec{x}) = \int_{-\infty}^{\infty} \max\left(y^* - y, 0\right) p(y|\vec{x}) \, \mathrm{d}y \tag{2.18}$$

There is no improvement if $y > y^*$.

$$= \int_{-\infty}^{y^*} (y^* - y) p(y|\vec{x}) \, \mathrm{d}y \tag{2.19}$$

$$= \int_{-\infty}^{y^*} (y^* - y) \underbrace{p(\vec{x}|y)}_{=\ell(\vec{x})} p(y) \frac{1}{p(\vec{x})} \, \mathrm{d}y \tag{2.20}$$

Used $p(x,y) = p(x|y)p(y) = p(y|x)p(x)$.

$$= \frac{\ell(\vec{x})}{\gamma \ell(\vec{x}) + (1-\gamma) g(\vec{x})} \underbrace{\int_{-\infty}^{y^*} (y^* - y) p(y) \, \mathrm{d}y}_{\text{independent of } \vec{x}} \tag{2.21}$$

Expanded $p(\vec{x})$ as $\int p(x|y)p(y)\,\mathrm{d}y$ and split the ℓ and g cases.

$$\propto \frac{1}{\gamma + (1-\gamma)\frac{g(\vec{x})}{\ell(\vec{x})}}\,, \tag{2.22}$$

where the term we neglected can be interpreted as the average improvement over all choices of $\vec{x}$, which we hope is large but which we cannot affect with a good choice of $\vec{x}$.

What this tells us is the rather intuitive fact that we want to select points which are favored by the "good" distribution ℓ and not by the "bad" distribution g, so that g/ℓ is small, which means that the expected improvement will be large. The formula also says that we should prefer smaller values of γ, but it does not take into account the fact that we need a reasonably large selection of points in order to properly model $\ell(\vec{x})$ without a large error. The balance among these two contrasting desires will dictate our choice of γ.

Once we are able to compute the expected improvement, the task is simple: we just need to generate a selection of points $\vec{x}_i$ from the distribution $\ell(\vec{x})$, evaluate their expected improvement (or just g/ℓ, really) and pick the one with the largest expected improvement as the one for which to evaluate the neural network (NN).

The one aspect still missing in the algorithm is how to estimate a probability distribution from a small set of samples, and how to calculate a new sample from it. There are different ways to do so; the one implemented in the TPE algorithm fits Gaussian Mixture Model to estimate $\ell(\vec{x})$ and $g(\vec{x})$ [BYC13, section 5].

3 MLGW-BNS

A The Stationary Phase Approximation

We can apply SPA to approximate the Fourier transform of an oscillatory function $f(t) = A(t)e^{-i\Phi(t)}$ as long as

1. $\Phi(t)$ is monotonic;
2. the evolution of the phase, $\dot{\Phi}$, is slowly changing: $\ddot{\Phi} \ll \dot{\Phi}^2$;
3. the amplitude $A(t)$ is slowly changing.

These seem like the right conditions intuitively, but are they correct?

If these conditions hold, then in the computation of the transform

$$\widetilde{f}(\omega) = \int_{-\infty}^{\infty} \mathrm{d}t\, A(t) e^{-i\Phi(t)+i\omega t} \tag{A.1}$$

the exponential term will be quickly oscillating, and therefore destructively interfering, almost everywhere; while near the time in which $\omega = \dot{\Phi}$ it will have a positive contribution.

With this in mind, we can approximate the integral: specifically, looking at the local integral contribution to the integral in figure 2 we can see that it approaches a Gaussian, which suggests expanding the argument of the exponential to second order around its stationary point,

$$-i\Phi(t) + i\omega t \approx \left(-i\Phi(t) + i\omega t\right)\Big|_{t_s} - \frac{i}{2}\,\ddot{\Phi}\Big|_{t_s} (t - t_s)^2\,, \tag{A.2}$$

where the stationary time t_s is precisely the one such that $\dot{\Phi}(t_s) = \omega$ (which means that the linear term of the expansion vanishes).

The Fourier integral then looks like

$$\widetilde{f}(\omega) \approx \int_{-\infty}^{\infty} \mathrm{d}t\, A(t) \exp\left(-i\Phi(t_s) + i\omega t_s\right) \exp\left(-\frac{i}{2}\ddot{\Phi}(t - t_s)^2\right), \tag{A.3}$$

so the t-dependent part of the integral looks like the product of the amplitude and a "complex Gaussian". There is an analytic formula for a complex Gaussian integral

$$\int e^{i\alpha x^2}\, \mathrm{d}x = \sqrt{\frac{\pi i}{\alpha}}\,, \tag{A.4}$$

however, in order to apply it, we need to approximate the amplitude as constant: $A(t) \approx A(t_s)$. This will be valid as long as the variation of the amplitude is small compared to the

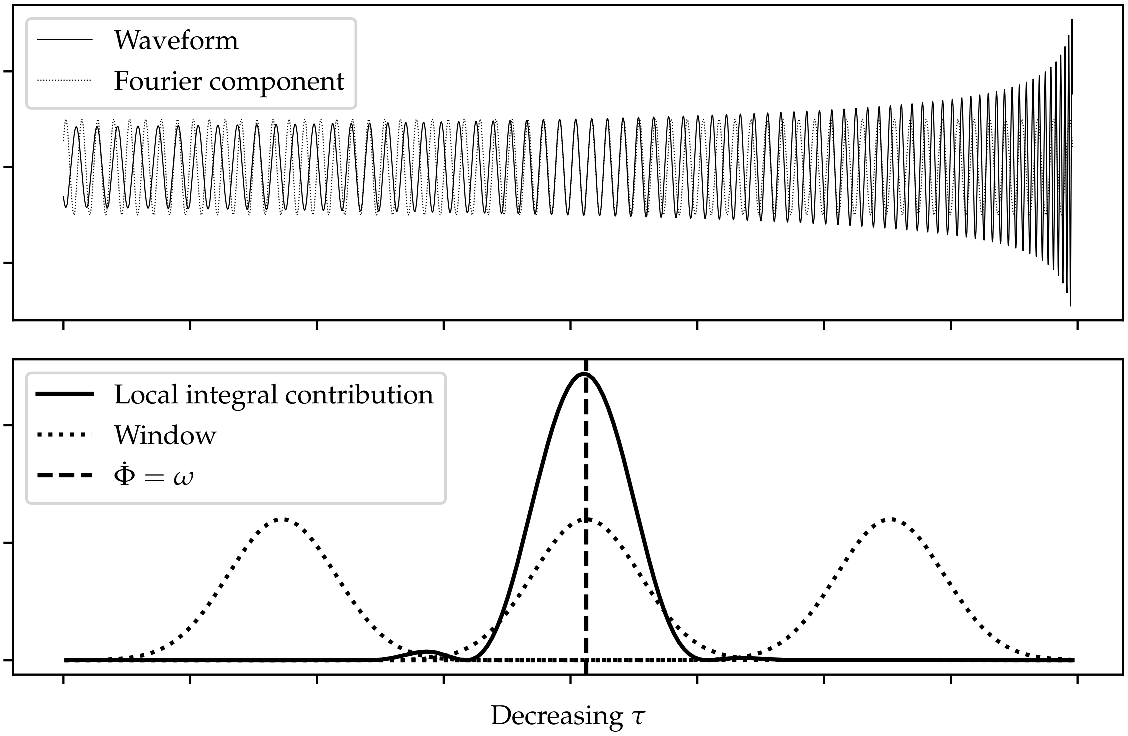

Figure 2: Visualization of the idea behind SPA. In the upper plot we show a typical waveform for a CBC together with a sinusoid corresponding to the ones used in the computation of the Fourier transform. The lower plot shows a windowed Fourier-like integral, in the form $\int h(t) \cos(\omega t) e^{-(t-t_0)^2/\sigma^2} \, \mathrm{d}t$ for some window width σ, as well as the window. It can be clearly seen that in the regions where there is a frequency mismatch the contribution vanishes.

width of the Gaussian we are integrating it against, which is $\sigma = 1/\sqrt{\ddot{\Phi}}$, so a condition we might write is $\dot{A}/A \lesssim 1/\sqrt{\ddot{\Phi}}$.

If this is the case, we can write the Fourier integral as

$$\widetilde{f}(\omega) \approx A(t_s) e^{-i\Phi(t_s) + i\omega t_s} \sqrt{\frac{-2\pi i}{\ddot{\Phi}}} \tag{A.5}$$

$$\left|\widetilde{f}(\omega)\right| \approx A(t_s(\omega)) \sqrt{\frac{2\pi}{\ddot{\Phi}(t_s(\omega))}} \tag{A.6}$$

$$\angle \widetilde{f}(\omega) \approx -\Phi(t_s(\omega)) + \omega t_s(\omega) - \frac{\pi}{4}\,, \tag{A.7}$$

where the $-\pi/4$ comes from the $\sqrt{-i}$.

We want to apply this to the specific case of A and Φ being given by polynomial expressions like the formulas describing quadrupole emission: (1.74) and (1.72); for simplicity, let us write these as

$$\Phi(t) = K_\Phi(-t)^\alpha \qquad \text{and} \qquad A(t) = K_A(-t)^\beta \,, \tag{A.8}$$

where the reason for writing $(-t)$ is that the aforementioned equations are written in terms of the time until merger τ — $t = -\tau$ is increasing and equal to 0 at the merger. We do not specify the polarization, the argument will apply for both. The true values for the exponents in the quadrupole approximation are $\alpha = 5/8$ and $\beta = -1/4$.

The first thing to do is to find $t_s(\omega)$: it is given by

$$\omega = \dot{\Phi} = \alpha K_\Phi (-t_s)^{\alpha-1} \tag{A.9}$$

$$t_S = -\left(\frac{\omega}{\alpha K_\Phi}\right)^{1/(\alpha-1)} . \tag{A.10}$$

The other quantities we need to compute are:

$$A(t_s) = K_A \left(\frac{\omega}{\alpha K_\Phi}\right)^{\beta/(\alpha-1)} \tag{A.11}$$

$$\Phi(t_s) = K_\Phi \left(\frac{\omega}{\alpha K_\Phi}\right)^{\alpha/(\alpha-1)} \tag{A.12}$$

$$\ddot{\Phi}(t_s) = K_\Phi \alpha(\alpha-1)(-t_s)^{\alpha-2} = -K_\Phi \alpha(\alpha-1)\left(\frac{\omega}{\alpha K_\Phi}\right)^{(\alpha-2)/(\alpha-1)} , \tag{A.13}$$

therefore the phase reads

$$\angle \widetilde{f}(\omega) = -K_\Phi \left(\frac{\omega}{\alpha K_\Phi}\right)^{\alpha/(\alpha-1)} - \omega \left(\frac{\omega}{\alpha K_\Phi}\right)^{1/\alpha} - \frac{\pi}{4} \tag{A.14}$$

$$= -\left(\frac{\omega^\alpha}{K_\Phi}\left(\alpha^{-\alpha} + \alpha^{-1}\right)\right)^{1/(\alpha-1)} - \frac{\pi}{4} , \tag{A.15}$$

while the amplitude reads

$$\left|\widetilde{f}(\omega)\right| = K_A \left(\frac{\omega}{\alpha K_\Phi}\right)^{\beta/(\alpha-1)} \sqrt{2\pi}\, \underbrace{\left(K_\Phi \alpha(\alpha-1)\left(\frac{\omega}{\alpha K_\Phi}\right)^{(\alpha-2)/(\alpha-1)}\right)^{-1/2}}_{1/\sqrt{\ddot{\Phi}}} \tag{A.16}$$

$$= K_A (\alpha K_\Phi)^{\frac{\beta-1/2}{\alpha-1}} \omega^{\frac{\beta-(\alpha-2)/2}{\alpha-1}} \sqrt{\frac{2\pi}{\alpha-1}} . \tag{A.17}$$

The last complication is the fact that the phase appears in the expression for the waveform (1.74) not with a complex exponential, but instead with a cosine or a sine: this is not a problem, we just need to use the relations $\cos x = (e^{ix} + e^{-ix})/2$ and $\sin x = (e^{ix} - e^{-ix})/2i$. The factor of i in the sine will correspond to a $\pi/2$ phase difference between the two polarizations, while the average of the two different exponentials will correspond to an amplitude reduction of a factor 2 if we restrict ourselves to positive frequencies.[12]

Substituting the values for α, β, K_Φ and K_A we get the following expressions [Mag07, eqs. 4.34–37], written in terms of $f = \omega/2\pi$:

$$\widetilde{h}_+(f) = \frac{1}{\pi^{2/3}}\sqrt{\frac{5}{24}}\frac{c}{r}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{5/6} f^{-7/6}\exp\left(\frac{3i}{4}\left(\frac{G\mathcal{M}_c}{c^3}8\pi f\right)^{-5/3} - i\frac{\pi}{4}\right)\left(\frac{1+\cos^2\iota}{2}\right) \tag{A.18}$$

$$\widetilde{h}_+(f) = \frac{1}{\pi^{2/3}}\sqrt{\frac{5}{24}}\frac{c}{r}\left(\frac{G\mathcal{M}_c}{c^3}\right)^{5/6} f^{-7/6}\exp\left(\frac{3i}{4}\left(\frac{G\mathcal{M}_c}{c^3}8\pi f\right)^{-5/3} + i\frac{\pi}{4}\right)\cos\iota\,. \tag{A.19}$$

The frequency dependence of the amplitude comes out to be $\left|\widetilde{f}(\omega)\right| \propto \omega^{-7/6} \propto f^{-7/6}$: it might be surprising to see that this is decreasing while the chirping waveform rises in amplitude! The reason for this fact is that each portion of the Fourier integral is weighted not only by the amplitude of the envelope $A(t)$ but also by the time the waveform "spends" in that frequency region, which decreases with time. These two effects compete, but $A \sim \tau^{-1/4}$ while the time spent in the frequency region is measured by $1/\sqrt{\ddot{\Phi}} \sim 1/\sqrt{\tau^{-11/8}} = \tau^{+11/16}$, so the latter wins: $\left|\widetilde{h}(f)\right| \sim \tau^{7/16} \propto \omega^{-7/6}$.

B Acronyms

GR general relativity

GW gravitational wave

EOB effective one-body

NR numerical relativity

PN post-Newtonian

PCA principal component analysis

BNS binary neutron star

BBH binary black hole

NS neutron star

BH black hole

SPA stationary phase approximation

[12] Since the Fourier transform is linear the two exponentials can be treated separately. The transform of a real signal is fully encoded by its positive-frequency components; in order for the integral not to vanish in the Gaussian approximation there needs to be a stationary point; the phase of the GW signal is strictly monotonic. These three assumptions lead to the fact that only one of the two exponentials will give a nonzero contribution while the other will have no stationary point, so only the division by 2 will matter in the average.

NN neural network

MOTPE multi-objective tree-structured Parzen estimator

TPE tree-structured Parzen estimator

`mlgw_bns` machine learning gravitational waves from binary neutron stars

TT transverse-traceless

CBC compact binary coalescence

References

[Abb+16] B. P. Abbott et al. "Observation of Gravitational Waves from a Binary Black Hole Merger". In: *Physical Review Letters* 116.6 (Feb. 11, 2016), p. 061102. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.116.061102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061102 (visited on 2021-04-04).

[Abb+17] B. P. Abbott et al. "GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral". In: *Physical Review Letters* 119.16 (Oct. 16, 2017), p. 161101. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.119.161101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.161101 (visited on 2021-04-04).

[Aki+19] Takuya Akiba et al. *Optuna: A Next-Generation Hyperparameter Optimization Framework*. July 25, 2019. arXiv: 1907.10902 [cs, stat]. URL: http://arxiv.org/abs/1907.10902 (visited on 2021-04-30).

[BYC13] J Bergstra, D Yamins, and D D Cox. "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: (2013), p. 9.

[Ber+11] James Bergstra et al. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011. URL: https://papers.nips.cc/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (visited on 2021-08-15).

[BGB21] Matteo Breschi, Rossella Gamba, and Sebastiano Bernuzzi. *${\tt Bajes}$: Bayesian Inference of Multimessenger Astrophysical Data, Methods and Application to Gravitational-Waves*. Jan. 29, 2021. arXiv: 2102.00017 [astro-ph, physics:gr-qc]. URL: http://arxiv.org/abs/2102.00017 (visited on 2021-05-03).

[Car19] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. New York: Cambridge University Press, 2019. 1 p. ISBN: 978-1-108-77038-5.

[Ein16] Albert Einstein. "Näherungsweise Integration Der Feldgleichungen Der Gravitation". In: *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin* (Jan. 1, 1916), pp. 688–696. URL: https://ui.adsabs.harvard.edu/abs/1916SPAW.......688E (visited on 2021-08-17).

[Ken07] Daniel Kennefick. *Traveling at the Speed of Thought*. 2007. ISBN: 978-0-691-11727-0. URL: https://press.princeton.edu/books/hardcover/9780691117270/traveling-at-the-speed-of-thought (visited on 2021-08-17).

[KB17] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 29, 2017. arXiv: 1412.6980 [cs]. URL: http://arxiv.org/abs/1412.6980 (visited on 2021-08-15).

[Knu92] Donald E. Knuth. *Two Notes on Notation*. Apr. 30, 1992. arXiv: math/9205211. URL: http://arxiv.org/abs/math/9205211 (visited on 2020-03-04).

[LL11] Ka-Wai Lo and Lap-Ming Lin. "THE SPIN PARAMETER OF UNIFORMLY ROTATING COMPACT STARS". In: *The Astrophysical Journal* 728.1 (Jan. 2011), p. 12. ISSN: 0004-637X. DOI: 10.1088/0004-637X/728/1/12. URL: https://doi.org/10.1088/0004-637x/728/1/12 (visited on 2021-08-20).

[Mag07] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments*. 1 edition. Oxford: Oxford University Press, Nov. 24, 2007. 576 pp. ISBN: 978-0-19-857074-5.

[Mag18] Michele Maggiore. *Gravitational Waves: Volume 2: Astrophysics and Cosmology*. Oxford: Oxford University Press, May 22, 2018. 848 pp. ISBN: 978-0-19-857089-9.

[MN17] Bryan F. J. Manly and Jorge A. Navarro Alberto. *Multivariate Statistical Methods: A Primer*. Fourth edition. Boca Raton: CRC Press, Taylor & Francis Group, 2017. 253 pp. ISBN: 978-1-4987-2896-6.

[Nie15] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. URL: http://neuralnetworksanddeeplearning.com (visited on 2021-03-23).

[Oza+20] Yoshihiko Ozaki et al. "Multiobjective Tree-Structured Parzen Estimator for Computationally Expensive Optimization Problems". In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. GECCO '20. New York, NY, USA: Association for Computing Machinery, June 25, 2020, pp. 533–541. ISBN: 978-1-4503-7128-5. DOI: 10.1145/3377930.3389817. URL: https://doi.org/10.1145/3377930.3389817 (visited on 2021-08-15).

[Pea01] Karl Pearson.
"LIII. On Lines and Planes of Closest Fit to Systems of Points in Space".
In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1, 1901), pp. 559–572. ISSN: 1941-5982.
DOI: 10.1080/14786440109462720.
URL: https://doi.org/10.1080/14786440109462720 (visited on 2021-08-13).

[Ped+11] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python".
In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL:
http://jmlr.org/papers/v12/pedregosa11a.html (visited on 2021-08-14).

[Rud16] Sebastian Ruder.
"An Overview of Gradient Descent Optimization Algorithms".
In: (Sept. 15, 2016).
URL: https://arxiv.org/abs/1609.04747v2 (visited on 2021-08-15).

[Sch+20] Stefano Schmidt et al.
Machine Learning Gravitational Waves from Binary Black Hole Mergers.
Nov. 3, 2020. arXiv: 2011.01958 [gr-qc].
URL: http://arxiv.org/abs/2011.01958 (visited on 2020-11-13).