TOPICS IN GRAVITATIONAL-WAVE ASTROPHYSICS

By

Alexander Harvey Nitz

Dissertation

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy in Physics

Syracuse University

August 2015

TOPICS IN GRAVITATIONAL-WAVE ASTROPHYSICS

By

Alexander Harvey Nitz

Dissertation

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy in Physics

Syracuse University

August 2015

Approved _____

Duncan A. Brown

Date _____

ABSTRACT

Contents

Preface

The work presented in this thesis stems from my participation in the LIGO Scientific Collaboration (LSC). This work does not reflect the scientific opinion of the LSC and it was not reviewed by the collaboration.

Acknowledgments

to

....

Chapter 1

Introduction

1.

Chapter 2

BNS SPIN

2.1 abstract

The detection of gravitational waves from binary neutron stars is a major goal of the gravitational-wave observatories Advanced LIGO and Advanced Virgo. Previous searches for binary neutron stars with LIGO and Virgo neglected the component stars' angular momentum (spin). We demonstrate that neglecting spin in matched-filter searches causes advanced detectors to lose more than 3% of the possible signal-to-noise ratio for 59% (6%) of sources, assuming that neutron star dimensionless spins, $c\mathbf{J}/GM^2$, are uniformly distributed with magnitudes between 0 and 0.4 (0.05) and that the neutron stars have isotropically distributed spin orientations. We present a new method for constructing template banks for gravitational wave searches for systems with spin. We present a new metric in a parameter space in which the template placement metric is globally flat. This new method can create template banks of signals with non-zero spins that are (anti-)aligned with the orbital angular momentum. We show that this search loses more than 3% of the maximium signal-to-noise for only 9% (0.2%) of BNS sources with dimensionless spins between 0 and 0.4 (0.05) and isotropic spin orientations. Use of this template bank will prevent selection bias in gravitational-wave searches and allow a more accurate exploration of the distribution of spins in binary neutron stars.

Topics in Gravitational-Wave Astrophysics

Alexander Harvey Nitz

August 4, 2015

2.2 Introduction

The second-generation gravitational wave detectors Advanced LIGO (aLIGO) and Advanced Virgo (AdV) [**?**, **?**] are expected to begin observations in 2015, and to reach full sensitivity by 2018-19. These detectors will observe a volume of the universe more than a thousand times greater than first-generation detectors and establish the new field of gravitational-wave astronomy. Estimated detection rates for aLIGO and AdV suggest that binary neutron stars (BNS) will be the most numerous source detected, with plausible rates of $\sim 40/\mathrm{yr}$ [**?**]. Gravitational wave observations of BNS systems will allow measurement of the properties of neutron stars and allow us to explore the processes of stellar evolution.

The gravitational waves that advanced detectors will observe from inspiralling BNS systems are well described by post-Newtonian theory [**?**]. As the neutron stars orbit each other, they lose energy to gravitational waves causing them to spiral together and eventually merge. If the angular momentum (spin) of the component neutron stars is zero, the gravitational waveform emitted depends at leading order on the chirp mass of the binary $\mathcal{M} = (m_1 m_2)^{3/5} / (m_1 + m_2)^{1/5}$ [**?**], where m_1, m_2 are the component masses of the two neutron stars, and at higher order on the symmetric mass ratio $\eta = m_1 m_2/(m_1 + m_2)^2$ [**?**, **?**, **?**, **?**, **?**, **?**]. If the neutron stars are rotating, coupling between the neutron stars' spin $\boldsymbol{S}_{1,2}$ and the orbital angular momentum $\boldsymbol{L}$ of the binary will affect the dynamics of BNS mergers [**?**, **?**, **?**, **?**]. We measure the neutron stars' spin using the dimensionless parameter $\boldsymbol{\chi}_{1,2} = \boldsymbol{S}_{1,2}/m_{1,2}^2$.

The maximum spin value for a wide class of neutron star equations of state is $\chi \equiv |\boldsymbol{\chi}| \sim 0.7$ [**?**]. However, the spins of neutron stars in BNS systems is likely

to be smaller than this limit. The spin period at the birth of a neutron star is thought to be in the range 10–140 ms [**?**, **?**]. During the evolution of the binary, accretion may increase the spin of one of the stars [**?**], however neutron stars are unlikely to have periods less than 1 ms [**?**], corresponding to a dimensionless spin of $\chi \sim 0.4$. The period of the fastest known pulsar in a double neutron star system, J0737–3039A, is 22.70 ms [**?**], corresponding to a spin of only $\chi \sim 0.05$. In this paper, we therefore consider two populations of neutron star binaries: the first has spins uniformly distributed from $\chi = 0$ to 0.4, the second, a sub-set of this, has spins between 0 and 0.05. This extended spin distribution allows for the possibility of serendipitous discovery of BNS systems in globular clusters, where the evolutionary paths may be different than that in field binaries [**?**]. Since supernova kicks may cause the direction of the neutron star's angular momentum to be misaligned with the orbital angular momentum of the binary [**?**], or the binaries may be formed by direct capture, we consider a population of binaries with an isotropic spin distribution.

Searches for binary neutron star systems in gravitational-wave detectors use template-based searches [**?**]. Data from the detector is correlated against a bank of known template waveforms, which cover the space of parameters searched over [**?**]. The template bank is constructed so that it covers the parameter space of interest so that any signal in this region will lose no more than 3% of the signal-to-noise ratio obtained by an exactly matching template. Alternative search methods have been proposed [**?**, **?**], however these still require the construction of a template bank to perform the search. The effect of spin-orbit and spin-spin interactions were neglected in previous BNS searches [**?**], as they do not have a significant effect on the $\sim$ 1600 gravitational wave cycles in the 40–2000 Hz sensitive band of first-generation detectors [**?**]. However, aLIGO and AdV will be sensitive to gravitational-wave frequencies between 10–2000Hz, increasing the number of cycles in band by an order of magnitude. Initial studies have demonstrated that over this band, the small secular effects produced by spin-orbit and spin-spin coupling will have a significant effect on the detectability of BNS systems with non-trivial component spins [**?**]. However, the current geometric method for placing BNS templates [**?**] does not incorporate spin. While numerical (stochastic) methods could be used to include spin, these require substantially more templates than a comparable geometric approach [**?**].

We present a new geometric algorithm for placing templates for BNS systems with

spin, which has a significantly higher sensitivity than previous searches. Our new algorithm constructs a metric on the parameter space using the various coefficients of the TaylorF2 expansion of the orbital phase as coordinates. In such a coordinate system the parameter space metric is globally flat, therefore we can transform into a Euclidean coordinate system. Finally, our method uses a Principal Coordinate Analysis to identify a two dimensional manifold that can be used to cover the aligned spin BNS parameter space using existing two dimensional lattice placement algorithms.

To demonstrate our new method, we first perform a systematic evaluation of the ability of a search that neglects spin to detect gravitational waves for BNS in aLIGO and AdV. We show that this search will lose more than 3% of the matched filter signal-to-noise ratio for 59% (6%) of signals if it is used to search for BNS systems with spins uniformly distributed between $0 \leq \chi_{1,2} \leq 0.4(0.05)$; this is unsatisfactory over a large region of the signal parameter space. We show that by considering BNS systems where the spin of the neutron stars are aligned with the orbital angular momentum (i.e. the binary is not precessing), we can create a two-dimensional template bank that is efficient at detecting spin-aligned BNS signals. Finally we demonstrate that this bank is sufficient to detect signals from generic spinning, precessing binaries in aLIGO and AdV. The spin-aligned bank loses more than 3% of the signal-to-noise ratio for only 9% (0.2%) of signals, sufficient to construct a sensitive and unbiased search for BNS systems in aLIGO and AdV.

2.3 BNS Search Sensitivity

We quantify the performance of templated matched-filter searches by the fitting factor (FF) of the search [**?**]. The fitting factor is the fraction of the signal-to-noise ratio that would be recovered when matching a given signal with the best matching waveform in the template bank. We first define the overlap between two templates h_1 and h_2 as

$$\mathcal{O}(h_1, h_2) = (\hat{h}_1|\hat{h}_2) = \frac{(h_1|h_2)}{\sqrt{(h_1|h_1)(h_2|h_2)}}. \tag{2.1}$$

which is defined in terms of the noise-weighted inner product [**?**]

$$(h_1|h_2) = 4\,\mathrm{Re}\int_0^\infty \frac{\tilde{h}_1(f)\tilde{h}_2^*(f)}{S_n(f)} df. \tag{2.2}$$

This overlap is the fraction of signal power that would be recovered by searching for the signal h_1 using a matched filter constructed from h_2. Maximizing the overlap over the time of arrival and waveform phase yields the match

$$\mathcal{M}(h_1, h_2) = \max_{\phi_c, t_c}(\hat{h}_1|\hat{h}_2(\phi_c, t_c)). \tag{2.3}$$

The mismatch, $1 - \mathcal{M}$, is the fraction of the optimal signal-to-noise ratio that is lost when searching for a signal h_1 with a template waveform h_2.

When searching for BNSs, we do not know the exact physical parameters of the system. We assume that the masses of the neutron stars lie between 1 and 3 $M_\odot$ and construct a bank of waveform templates $\{h_b\}$ to span this region of the mass parameter space. To measure the sensitivity of this bank to a gravitational waveform h_s with unknown parameters, we compute the fitting factor

$$\mathrm{FF}(h_s) = \max_{h \in \{h_b\}} \mathcal{M}(h_s, h), \tag{2.4}$$

where we have maximized the match over all the templates in the bank. In searches for gravitational waves using LIGO and Virgo, the bank is constructed such that the fitting factor for any signal in the target parameter space will never be less than 0.97. At least one of the templates in the bank must have a maximized overlap of 0.97 (or more) with the signal. This value is chosen to correspond to an event rate loss of no more than 10% of possible sources within the range of the detectors [**?**]. In this paper, we use a fitting factor of 0.97 to construct search template banks.

We now test whether a bank of templates that does not model the effect of spin is sufficient to detect generic, spinning BNS sources in aLIGO and AdV. We create a bank of non-spinning templates that would recover any non-spinning BNS system with a fitting factor greater than 0.97. This bank is constructed using TaylorF2 waveforms, which are constructed using the stationary phase approximation to the gravitational-wave phasing accurate to 3.5 post-Newtonian (PN) order [**?**, **?**]. To create a bank of these waveforms we use the hexagonal-placement method defined in [**?**], which was used in the majority of previous searches in LIGO and Virgo [**?**, **?**, **?**]. This template bank is placed using the metric given in [**?**], which is valid, by construction, for templates at 2PN order. Our signal waveforms are constructed

using the SpinTaylorT4 waveform [**?**], a time-domain waveform accurate to 3.5PN order in the orbital phase which includes the leading order spin-orbit, spin-spin, and precessional modulation effects and implemented in the LSC Algorithm Library Suite [**?**]. We first confirm that although the bank is constructed at 2PN order, it yields fitting factors greater than 0.97 for both the TaylorF2 and SpinTaylorT4 non-spinning waveforms at 3.5PN order. To simulate a population of spinning BNS sources, we generate 100,000 signals with component masses uniformly distributed between 1 and 3 $M_{\odot}$ and dimensionless spin magnitudes uniformly distributed between 0 and 0.4. The orientation of the spin, the orientation of the orbital angular momentum, and the sky location are isotropically distributed. To model the sensitivity of a second generation gravitational wave interferometer, we use the aLIGO zero-detuned, high-power sensitivity curve [**?**]. For our simulations, we use a lower frequency cutoff of 15Hz.

We note that for non-precessing systems the fitting factor is independent of the detector alignment and location; however this statement is not true for precessing systems. For such systems, however, the distribution of fitting factors over a population of sources will be independent of the detector alignment and location. Therefore, for this study we calculate the fitting-factor for a single detector with an arbitrary location and position.

In Fig. 1 we show the distribution of fitting factors obtained when searching for our population of BNS sources with the non-spinning template bank. We see that 59% of signals were recovered with a fitting factor less than 0.97. If the maximum spin magnitude is restricted to 0.05, we find that 6% of signals are recovered with a FF less than 0.97. If BNS systems do exist with spin magnitudes up to 0.4, a template bank that captures the effects of spin will be required to maximize the number of BNS detections. Detection efficiency will be greatly reduced by using a template bank that only contains waveforms with no spin effects. Even under the assumption that component spins in BNS systems will be no greater than 0.05, detection efficiency will be decreased if the effect of spin on the signal waveform is ignored.

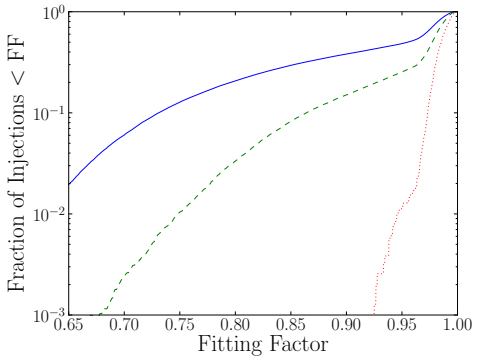

Figure 1: The distribution of fitting factors obtained by searching for the precessing BNS systems described in section 2.3 with component spins up to 0.4 (blue solid line), 0.2 (green dashed line), and 0.05 (red dotted line) using the non-spinning BNS template bank described in section 2.3 and the advanced LIGO, zero-detuned, high-power PSD with a 15Hz lower frequency cutoff.

2.4 A template placement algorithm for aligned-spin BNS templates

As we have demonstrated in the previous section, there is a substantial region of the BNS parameter space where a significant loss in signal-to-noise ratio would be encountered when searching for astrophysically plausible, spinning BNS systems with non-spinning templates. It has been suggested that using BNS templates where the spins of the system are aligned with the orbital angular momentum is sufficient for detecting generic BNS systems with second-generation detectors [**?**] using TaylorF2 templates that incorporate the leading order spin-orbit and spin-spin corrections [**?**].

In this section we use these spin-aligned waveforms to construct a template bank that attempts to cover the full space of astrophysically plausible BNS spin configurations. This template bank should contain as few templates as possible, while still being able to detect any BNS system that might be observed with aLIGO and AdV. To achieve this, it is important to assess the "effective dimension" of the space, which is defined as the number of orthogonal directions over which template waveforms need to placed in order to cover the full physically possible parameter range. We demonstrate that the effective dimension of this parameter space is only two dimensional. For BNS systems in aLIGO and AdV the extent of the physical parameter space in the remaining directions is smaller than the coverage radius of a template and can be

neglected.

As the effective dimension of the space is two-dimensional, a hexagonal placement algorithm, similar to that used in previous searches of LIGO and Virgo data, could be employed to cover the space. This allows our new method to be incorporated into existing search pipelines in a straightforward way.

Since BNS systems coalesce at $\sim$ 1500 Hz, significantly higher than the most sensitive band of the detectors, the waveform will be dominated by the inspiral part of the signal [**?**]. The effect of component spin on BNS inspiral waveforms has been well explored in the literature [**?**, **?**, **?**, **?**]). For spin-aligned (i.e. non-precessing) waveforms, the dominant effects of component spin are spin-orbit coupling, which enters the waveform phasing at 1.5PN order, and spin1-spin2 coupling, which enters the waveform phasing at 2PN order. Other spin-related corrections to the PN phasing have been computed [**?**, **?**], however, in this work we mainly restrict to only the two dominant terms. The methods described here are easily extendable to include additional spin correction terms and this does not significantly change our results, as we demonstrate at the end of this section.

To construct a bank to search for generic BNS signals, we use TaylorF2 waveforms accurate to 3.5PN order in orbital phase and including the leading order spin-orbit and spin-spin terms given by [**?**, **?**]

$$\tilde{h}(f) = A(f;\theta_x)e^{i\Psi(f;\lambda_i)} \tag{2.5}$$

where θ_x describe the various orientation angles that only affect the amplitude and overall phase of the observed gravitational waveform [**?**]. The phase Ψ is given by

$$\Psi = 2\pi f_0 x t_c - \phi_c + \lambda_0 x^{-5/3} + \lambda_2 x^{-1} + \lambda_3 x^{-2/3} + \lambda_4 x^{-1/3} + \lambda_{5L}\log(x) + \lambda_6 x^{1/3} + \lambda_{6L}\log(x)x^{1/3} + \lambda_7 x^{2/3}, \tag{2.6}$$

where f is the frequency, f_0 is a fiducial frequency, $x = f/f_0$, t_c is the coalescence

time, ϕ_c is a constant phase offset. The PN phasing terms are

$$\lambda_0 = \frac{3}{128}(\pi\mathcal{M}f_0)^{-5/3}, \tag{2.7}$$

$$\lambda_2 = \frac{5}{96\eta^{2/5}}\left(\frac{743}{336}+\frac{11}{4}\eta\right)(\pi\mathcal{M}f_0)^{-1}, \tag{2.8}$$

$$\lambda_3 = -\frac{3\pi}{8\eta^{3/5}}\left(1-\frac{1}{4\pi}\beta\right)(\pi\mathcal{M}f_0)^{-2/3}, \tag{2.9}$$

$$\lambda_4 = \frac{15}{64\eta^{4/5}}\left(\frac{3058673}{1016064}+\frac{5429}{1008}\eta+\frac{617}{144}\eta^2-\sigma\right)(\pi\mathcal{M}f_0)^{-1/3} \tag{2.10}$$

$$\lambda_{5L} = \frac{3}{128\eta}\left(\frac{38645\pi}{756}-\frac{65\pi}{9}\eta\right) \tag{2.11}$$

$$\lambda_6 = \frac{3}{128\eta^{6/5}}\left(\frac{11583231236531}{4694215680}-\frac{640\pi^2}{3}-\frac{6848}{21}\left(\gamma_E+\log 4-\frac{1}{5}\log\eta+\frac{1}{3}\log(\pi\mathcal{M}f_0)\right)\right.$$
$$\left.-\frac{15737765635}{3048192}\eta+\frac{2255\pi^2}{12}\eta+\frac{76055}{1728}\eta^2-\frac{127825}{1296}\eta^3\right)(\pi\mathcal{M}f_0)^{1/3} \tag{2.12}$$

$$\lambda_{6L} = -\frac{1}{128\eta^{6/5}}\frac{6848}{21}(\pi\mathcal{M}f_0)^{1/3} \tag{2.13}$$

$$\lambda_7 = \frac{3}{128\eta^{7/5}}\left(\frac{77096675\pi}{254016}+\frac{378515\pi}{1512}\eta-\frac{74045\pi}{756}\eta^2\right)(\pi\mathcal{M}f_0)^{2/3}, \tag{2.14}$$

where γ_E is the Euler gamma constant, β (the dominant spin-orbit coupling term) and σ (the dominant spin-spin coupling term) are given by

$$\beta = \frac{1}{12}\sum_{i=1}^{2}\left[113\left(\frac{m_i}{m_1+m_2}\right)^2+75\eta\right]\hat{\boldsymbol{L}}\cdot\boldsymbol{\chi}_i \tag{2.15}$$

$$\sigma = \frac{\eta}{48}\left(-247\boldsymbol{\chi}_1\cdot\boldsymbol{\chi}_2+721\hat{\boldsymbol{L}}\cdot\boldsymbol{\chi}_1\hat{\boldsymbol{L}}\cdot\boldsymbol{\chi}_2\right). \tag{2.16}$$

and $\hat{\boldsymbol{L}}$ is the unit vector in the direction of the orbital angular momentum. Note that above we have omitted the λ_5 term, as it has no dependance on frequency and is therefore included in the constant phase offset, ϕ_c.

Our goal is to construct a template bank containing the minimum number of waveforms for which any plausible BNS signal has a FF of 0.97 or higher. To place a template bank, we follow the method of Owen [**?**]. We first construct a metric on the waveform parameter space that describes the mismatch between infinitesimally

separated points,

$$\mathcal{O}(h(\boldsymbol{\theta}), h(\boldsymbol{\theta} + \delta\boldsymbol{\theta})) = 1 - \sum_{ij} g_{ij}(\boldsymbol{\theta})\, \delta\theta^i\, \delta\theta^j, \tag{2.17}$$

with the metric given by,

$$g_{ij}(\boldsymbol{\theta}) = -\frac{1}{2}\frac{\partial^2 \mathcal{O}}{\partial\delta\theta^i \partial\delta\theta^j} = \left(\frac{\partial h(\boldsymbol{\theta})}{\partial\theta^i}\middle|\frac{\partial h(\boldsymbol{\theta})}{\partial\theta^j}\right) \tag{2.18}$$

and where $\boldsymbol{\theta}$ describes the parameters of the signal, in this case the masses and the spins.

This metric is used to approximate the mismatch in the neighborhood of any point. When doing this care must be taken to choose a "good" set of coordinates where extrinsic curvature is minimized. If a "bad" set of coordinates is chosen, the region in which this approximation can be used will be very small. To minimize this issue when placing the two-dimensional non-spinning bank, the masses m_1, m_2 are transformed into the "chirp times" τ_0, τ_3 [**?**]. In this coordinate system, ellipses are constructed that describe fitting factors greater than 0.97 around a point and hexagonal placement is used to efficiently tile the space to achieve the desired minimal match [**?**].

To construct our new bank, we treat the six λ_i and two λ_{iL} components, given in Eq. (2.7), as eight independent parameters, as in [**?**]. The range of possible physical values will trace out a four-dimensional manifold in the eight dimensional parameter space given by the λ_α, where α is an index that takes both i and iL values. We will demonstrate that this eight-dimensional parameter space allows us to construct a metric without intrinsic curvature.

As shown in [**?**] it is possible to evaluate the derivative in (4.21), maximizing over the phase, ϕ_C, to give the metric in terms of a 9 dimensional space:

$$\gamma_{\alpha\beta} = \frac{1}{2}\left(\mathcal{J}[\psi_\alpha\psi_\beta] - \mathcal{J}[\psi_\alpha]\mathcal{J}[\psi_\beta]\right) \ . \tag{2.19}$$

In this expression ψ_α is given by

$$\psi_0 = \frac{\partial\Psi}{\partial t_c} = 2\pi f_0 x \tag{2.20}$$

$$\psi_i = \frac{\partial\Psi}{\partial\lambda_i} = x^{(i-5)/3} \tag{2.21}$$

$$\psi_{iL} = \frac{\partial\Psi}{\partial\lambda_{iL}} = x^{(i-5)/3}\log(x) \tag{2.22}$$

and $\mathcal{J}$ is the moment functional of the noise PSD [**?**, **?**]

$$\mathcal{J}[a(x)] = \frac{1}{I(7)}\int_{x_L}^{x_U}\frac{a(x)x^{-7/3}}{S_h(xf_0)}\mathrm{d}x \ , \tag{2.23}$$

where

$$I(q) \equiv \int_{x_L}^{x_U}\frac{x^{-q/3}}{S_h(xf_0)}\mathrm{d}x \tag{2.24}$$

and x_U and x_L correspond to the lower and upper bounds of frequency in the integral. Unless stated otherwise we use $f_L = x_L f_0 = 15$Hz for the aLIGO PSD and choose 2000Hz for the upper frequency cutoff, $f_U = x_U f_0$. While it is unphysical to use the same upper frequency cutoff for all systems, especially as we are not including a merger in our waveform model, it is necessary to make this assumption to ensure that our metric will be flat. For BNS systems this approximation is fair to use as such systems will merge at frequencies that are outside the sensitive range of the advanced detectors and thus our calculation of signal power is not affected by assuming that all BNS systems merge abruptly at 2000Hz. This approach was also used in [**?**] for computational efficiency.

Following [**?**] we can then maximize this expression over t_C to give the metric in terms of the eight λ_α

$$g_{\alpha\beta} = \gamma_{\alpha\beta} - \frac{\gamma_{0\alpha}\gamma_{0\beta}}{\gamma_{00}}. \tag{2.25}$$

It is worth highlighting that the parameter space metric $g_{\alpha\beta}$, in the λ_α coordinate system, has no dependence on the values of λ_α. In other words, the parameter space is globally flat in this eight-dimensional parameter space.

Although this eight-dimensional metric is globally flat, we have increased the dimensionality of the physical waveform space by a factor of two. However, we can transform this metric to a new coordinate system that will allow us to assess the

effective dimensionality of the parameter space.We first rotate and rescale the metric to transform to a Cartesian coordinate system. We now use indicies i, j to number the remaining eight λ_α coordinates. As g_{ij} is a real, symmetric matrix we can use the eigenvalues and eigenvectors of the metric to rotate into an orthonormal coordinate system defined by

$$\mu_i = \sum_j \left(V_{ij} \sqrt{E_i} \right) \lambda^j, \tag{2.26}$$

where V_{ij} describes the eigenvectors of g_{ij} and E_i its corresponding eigenvalues. We use the convention that V_{ij} is the j^{th} component of the i^{th} eigenvector, and the eigenvectors are normalized by $V^T V = \mathcal{I}$. In this coordinate system, the metric, g'_{ij}, will be the identity matrix. Next, we perform a rotation to align the axis of the parameter space with the principal components of the physically possible region of the space. The physically allowed ranges of the masses and spins cover only a limited region in the parameter space. The extent of the physically relevant region of the space in a certain direction may be thin relative to the desired mismatch. By orienting the coordinate system along the principal directions we can easily identify any orthogonal directions in which the physical region is sufficiently thin that we do not need to place templates in those directions. This will allow us to assess the effective dimension of the parameter space, or, in other words, how many directions we need to consider when placing a template bank. Transforming to a Cartesian coordinate system also helps with template placement, as it is trivial to place templates using the optimal A_n^* lattice [**?**] in a 2, 3 or 4 dimension Cartesian coordinate system

To perform the second rotation we make use of the fact that in a Cartesian coordinate system we are free to rotate the coordinates without changing the form of the metric. We would like to rotate the coordinates so that the greatest extent of the template bank lies along as few directions as possible. To accomplish this we first draw many examples of physical parameters of the masses and spins, and calculate the corresponding values of μ_i for each of these points. We then do a Principal Component Analysis on this dataset, which amounts to finding the eigenvectors of the covariance matrix from the set of μ_i. This produces a rotation into a new set of coordinates given by

$$\xi_i = \sum_j \left(C_{ij} \mu^j \right), \tag{2.27}$$

where C_{ij} contains the eigenvectors of the covariance matrix using the same conventions as for V_{ij}. The rotation of course leaves the metric Cartesian, but now the bank is oriented along the principal axes and it is much easier to visualize the shape of the boundaries and determine how to perform the template placement.

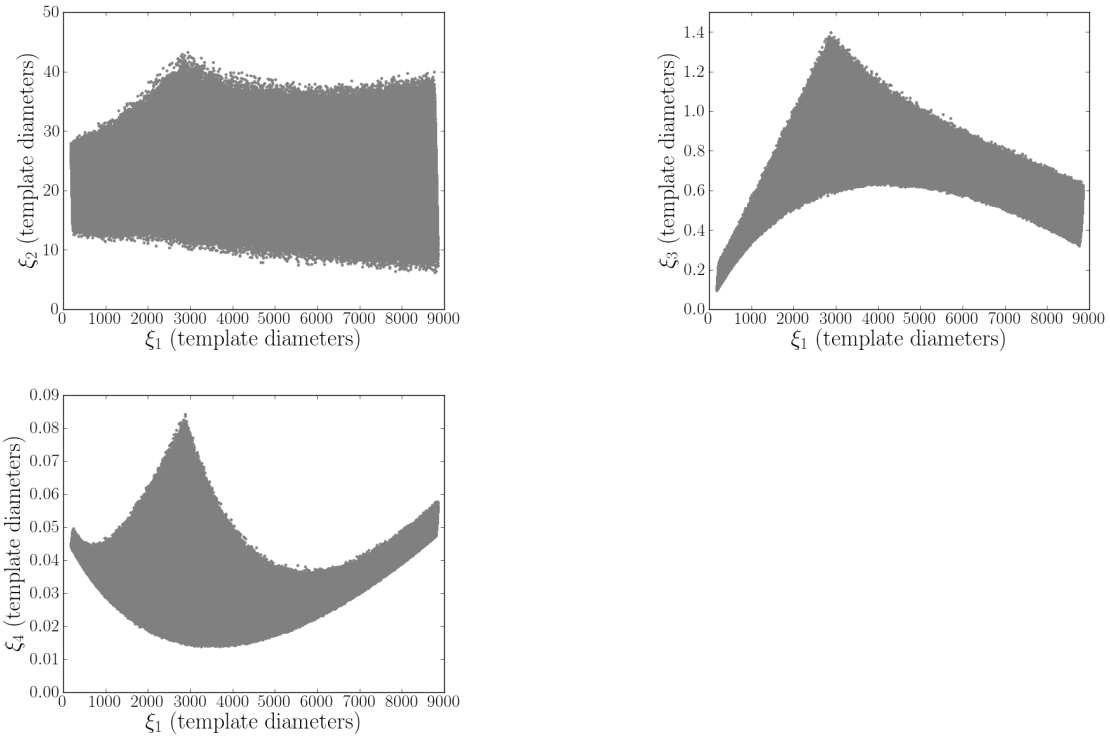

Figure 2: The extent of the binary neutron star, $\chi_i < 0.4$, parameter space in the ξ_2, ξ_3 and ξ_4 directions, plotted against ξ_1. The ξ_i coordinates have been scaled such that one unit corresponds to the coverage diameter of a template at 0.97 mismatch. Generated using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

We now use this method to construct a template bank where the spin of each component neutron star is restricted to 0.4. When this metric is constructed using the aLIGO, zero-detuned high-power noise curve with a lower frequency cut-off of 15Hz we show that, although many additional templates are required to cover an aligned spinning parameter space when compared to the non spinning space, the effective dimension for these BNS systems is still two.

We begin by attempting to visualize the space. We will refer to ξ_1 as the direction along which the parameter space has the biggest extent (the dominant direction) and

ξ_8 as the direction with the smallest extent (the least-dominant direction). We draw a large set of points, with random values of masses and spins, and transform these points into the ξ_i coordinate system. The position of these points is shown in Figure 2, where we plot the extent of $\xi_{2,3,4}$ against ξ_1.

In Figure 2 and subsequent plots, we have scaled the ξ_i direction such that one unit corresponds to the coverage diameter of a template at 0.97 mismatch. Equivalently, we have scaled the directions such that two points separated by 0.5 units (one template radius) in any direction have a match of 0.97[1]. We remind that mismatch is proportional to distance squared and therefore two points separated by one unit would have a match of 0.88

Immediately we notice that the extent along the ξ_4 direction is small compared to the diameter of a template. We can also see that the extent along the ξ_3 direction is comparable to a template diameter, while the ξ_1 and ξ_2 directions have much larger extents and clearly need to be gridded over. The extent in the other 4 directions is smaller than ξ_4 and can be completely ignored. This hierarchy of measurable parameters may be a generic feature according to the model of [**?**].

The plot of ξ_1 against ξ_3 in Figure 2 can be somewhat misleading as we have projected out the ξ_2 direction. It is more informative to investigate the depth of ξ_3 at fixed values of ξ_1 and ξ_2 and translate this into the maximum mismatch that would be obtained if one were to assume that there is no width in the third direction. In Figure 3 we show the maximum mismatch between the central and extremal values of the possible range of ξ_3 (and ξ_4) as a function of the two primary directions. This is calculated by binning the points mentioned above into bins in ξ_1 and ξ_2, where the bin width is equal to one template radius. We then determine the extremal values of ξ_3 (and ξ_4) for the points in each bin. From Figure 3 we can see that, while there are small areas of parameter space where up to a 1.6% loss in SNR would be incurred from assuming the ξ_3 direction had no depth, most areas of the parameter space are very thin in the ξ_3 direction. This figure also helps to reinforce the fact that the depth in the fourth direction is negligible, as, even in the worst region of the space, no more than 0.01% of SNR would be lost by assuming ξ_4 had no depth. The depth of the $\xi_{5,8}$ directions are even smaller than ξ_4.

In this coordinate system it is easy to explore how the size of the parameter

[1]The unscaled distance between two points with a match of 0.97 would be $(1 - 0.97)^{0.5} = 0.17$

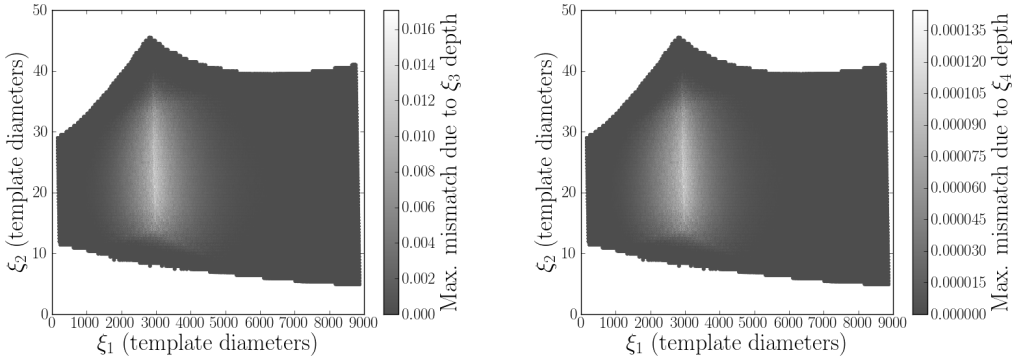

Figure 3: The mismatch between the edge and centre of the physically possible range of ξ_3 (top) and ξ_4 (bottom) values as a function of ξ_1 and ξ_2. The ξ_i coordinates have been scaled such that one unit corresponds to the coverage diameter of a template at 0.97 mismatch. Plotted for a binary neutron star parameter space with spins restricted to 0.4 using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

space depends on the maximal spins of the component neutron stars. In Figure 4 we show the extent of the physical space for aligned spinning BNS systems, with maximum component spins of 0.4, 0.2 and 0.1, compared to that of non-spinning systems. Ignoring any issues related to the depth of the ξ_3 direction, one can clearly see that to cover the aligned spin parameter space will require a great deal more templates than the non spinning parameter space.

From these results we can see that a 2 dimensional template bank would be sufficient to cover the aligned spin parameter space for BNS systems in the advanced detector era. Specifically, we would advocate placing a hexagonal lattice in the ξ_1, ξ_2 coordinates and setting the value of $\xi_{3..8}$ to be the middle of the possible range of those parameters at the given position of ξ_1, ξ_2. For the regions of parameter space where the depth of ξ_3 is not negligible, one could either ignore it, understanding that the resulting bank will not have a fitting factor of 0.97 in this region. Alternatively, one could stack templates in the region where ξ_3 is deepest to minimize this effect.

For this work we chose to employ a hexagonal template bank in the ξ_1, ξ_2 coordinates, stacking the templates in the ξ_3 direction, where necessary, to ensure that the maximum mismatch due to the depth of ξ_3 is no more than 0.25%. For an aligned-spin template bank where the spin of each component is restricted to 0.4, using the advanced LIGO, zero-detuned high-power noise curve with a lower frequency cut-off

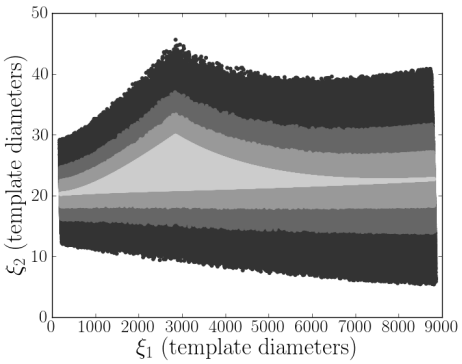

Figure 4: The size of the BNS parameter space as a function of the maximum spin. The darkest points indicate points with spin on both components constrained to 0.4, then, in order of increasing lightness, we show points constrained to a maximal spin of 0.2 and 0.1, finally the lightest points show points with no spin. The ξ_i coordinates have been scaled such that one unit corresponds to the coverage diameter of a template at 0.97 mismatch. This plot was generated using the zero-detuned, high-power aLIGO sensitivity curve with a 15Hz lower frequency cut off.

of 15Hz, we find that approximately 520,000 templates are required. Roughly 100,000 of these templates were added by the stacking process.

We can verify that the template bank algorithm is working correctly by repeating the simulation described in section 2.3, but evaluating the fitting factor between our bank of aligned-spin template waveforms and a set of signals that is restricted to having spins that are (anti-)aligned with the orbital angular momentum. The results of this simulation are shown in figure 5 and one can see that with our bank we do not observe fitting factors lower than 0.97 when searching for aligned spin BNS systems.

In the previous paragraphs we have restricted attention to the aLIGO zero-detuned, high-power predicted sensitivity with a 15Hz lower frequency cut off. However, we should verify that the conclusions we have drawn are valid for AdV, whose PSD is different from that of aLIGO, as shown in Figure 6. Additionally we should also show that the choice to use a 15Hz cut off in the aLIGO PSD does not affect the conclusions made in this section.

The process we described above is applicable for any PSD, and therefore we can use it directly to determine the ξ_i directions for the AdV PSD, or the aLIGO PSD with a 10Hz lower frequency cutoff. In Figure 7 we plot ξ_1 against ξ_2 for both PSDs while

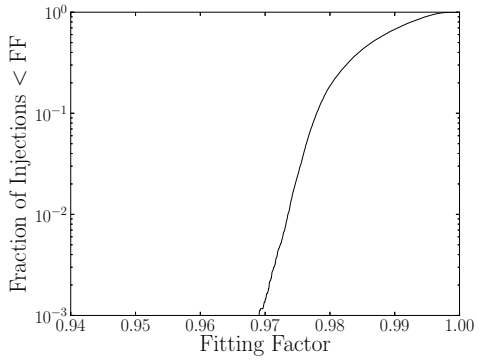

Figure 5: The distribution of fitting factors obtained by searching for aligned spin, binary neutron star systems, with spin magnitudes restricted to 0.4 using the aligned-spin BNS template bank described in section 2.4 and the aLIGO, zero-detuned, high-power PSD with a 15Hz lower frequency cutoff.

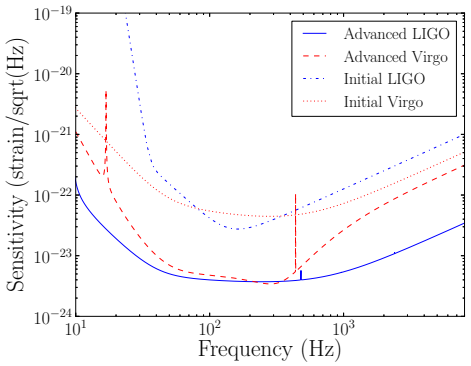

Figure 6: The amplitude spectral density for the aLIGO zero-detuned high-power design sensitivity (blue solid curve), AdV design sensitivity (red dashed curve), initial LIGO design sensitivity (blue bot-dash curve) and initial Virgo design sensitivity (red dotted curve).

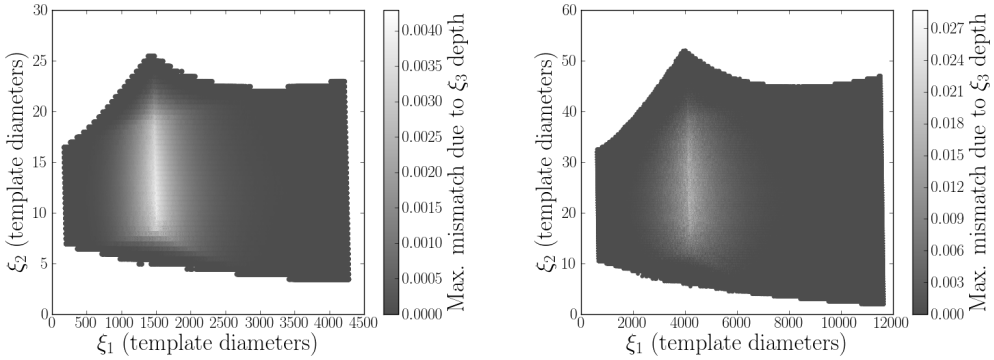

Figure 7: The mismatch between the edge and centre of the third dominant direction as a function of the first and second dominant directions when using the Virgo noise curve (top) and when using the advanced LIGO noise curve with a 10Hz lower frequency cut off (bottom). The ξ_i coordinates have been scaled such that one unit corresponds to the coverage diameter of a template at 0.97 mismatch. Plotted for a binary neutron star parameter space with spins restricted to 0.4.

the color shows the mismatch between the center and edges in the ξ_3 direction. This plot can be directly compared to Figure 3. We notice that the size of the parameter space for the AdV PSD is significantly smaller than for the aLIGO PSD in all 3 of the dominant directions. Therefore our conclusions for aLIGO are still valid for AdV. Using our method we find that we require approximately 120,000 templates to cover the parameter space for AdV, in comparison to approximately 520,000 templates for aLIGO.

By comparing the results when using the aLIGO PSD with a 10Hz and 15Hz lower cut off we observe that using a 10Hz lower frequency cut off will increase the number of necessary templates from ~ 520000 to ~ 860000. However the shape of the parameter space, and thus our final conclusions, are unaffected when using a 10Hz lower frequency cutoff. However, in this case we see larger mismatches due to the depth of ξ_3 and therefore the process of stacking templates is important when using a 10Hz lower cut off. However, even in this case, we do not feel that the depth is large enough everywhere in the space to justify using a fully 3-dimensional placement algorithm.

Finally, we wish to investigate the effect that the higher order spin contributions to the orbital phase have on our method. To do this we repeat the process described above, but include the spin(1)-spin(1) and spin(2)-spin(2) contributions to the σ term

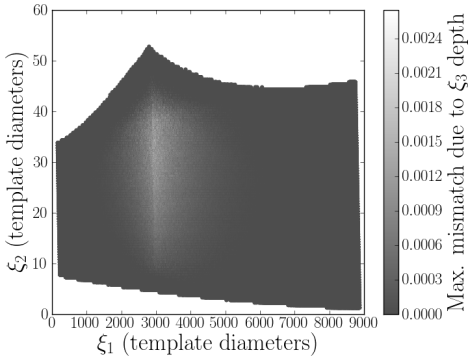

Figure 8: The mismatch between the edge and centre of the third dominant direction as a function of the first and second dominant directions using waveforms incorporating the sub-dominant spin corrections to the orbital phase. The ξ_i coordinates have been scaled such that one unit corresponds to the coverage diameter of a template at 0.97 mismatch. Plotted for a binary neutron star parameter space with spins restricted to 0.4 using the zero-detuned, high-power aLIGO sensitivity curve with a 15Hz lower frequency cut off.

at 2PN order and also the 2.5PN spin-orbit term as given in [**?**]. In Figure 8 we plot ξ_1 against ξ_2 when these higher order spin terms are included, the color shows the mismatch between the center and edges in the ξ_3 direction. This plot can be directly compared to Figure 3. By comparing these plots we can see that including the higher order spin terms has caused the parameter space to have a larger extent in the ξ_2 direction. However, the depth of the space in the ξ_3 direction has reduced by almost an order of magnitude. In this case the stacking process is not required and the resulting bank consists of ~ 560000 templates.

2.5 Comparison to alternative placement methods

An alternative approach to template placement for aligned spin systems is to use templates with "unphysical" values of the symmetric mass ratio, η. That is, to use non-spinning templates, with the desired range of chirp mass but where the range of η values is extended to include both values of η that are much lower than the relevant parameter space and values of η that are much higher, including templates with η greater than the physically possible limit of 0.25.

We can understand this unphysical η approach in terms of our ξ_i coordinate system

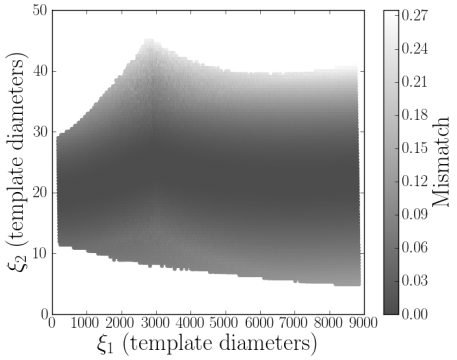

Figure 9: The mismatch between unphysical η and aligned spin BNS templates as a function of the first and second dominant directions. In making this plot we assume that ξ_3 has no depth in the aligned spin case by taking the central value where ξ_3 has a range of values. This plot was generated with spins restricted to 0.4 using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

by noting that it is always possible to produce a template with any possible value of ξ_1 and ξ_2 that is within the BNS parameter space, by using non-spinning templates with unrestricted values of η. By generating a set of templates in the ξ_1, ξ_2 directions, where we restrict the chirp mass to be that possible for BNS systems, but where η ranges from 0.1 to 0.7 we are able to cover the full physically possible space in ξ_1, ξ_2. However, the disadvantage to using unphysical η templates is that the points will not take the correct values of ξ_3. The colorbar on Figure 9 indicates the mismatch between unphysical η templates and aligned-spin templates as a function of ξ_1 and ξ_2. In making this plot we assume that ξ_3 has no depth in the aligned spin case by taking the central value where ξ_3 has a range of values.

While unphysical η templates will produce an increase in efficiency when compared with non-spinning templates, the method is not as efficient as the aligned spin geometrical placement we have described. In addition, both methods require the same number of templates to cover the parameter space. Therefore, we would recommend using aligned spin templates placed using our metric algorithm as opposed to unphysical η templates.

Finally, we wish to compare the performance of this geometrical algorithm with the stochastic bank proposed in [**?**, **?**]. The stochastic placement works by randomly placing points within the parameter space and rejecting points that are too "close" to

points already in the bank. This has the advantage that it is valid for any parameter space metric, so we could use any of the metrics discussed above. However, it is more computationally efficient to use the Cartesian ξ_i or μ_i coordinate system rather than the non-Cartesian metric given above.

The disadvantage to a stochastic bank, when compared to a geometrically placed bank, is that it will require more templates to achieve the same level of coverage [**?**, **?**]. For our parameter space, consisting of BNS signals with component spins up to 0.4 and using the advanced LIGO zero-detuned high-power design curve with a 15Hz lower frequency cut-off, we found that the stochastic placement produced a bank containing $\sim$ 750000 templates, which is 44% more than with the geometrical placement. However, stochastic placement can still be used to place templates when no analytical metric is known, such as when the merger becomes important. In such regions of parameter space, the stochastic placement may still be the best algorithm to use to place a template bank.

2.6 Performance of the aligned spin template bank

In this section we would like to investigate the improvement in the detection of generic BNS systems that results from using a template bank that includes the dominant, non-precessing, spin effects. To do this we use the aligned spinning bank that we detail in section 2.4 and compare this to the results of using a nonspinning bank as shown in section 2.3.

Using our aligned spin template bank, we repeat the investigation from section 2.3. We create a population of source BNS signals identical to those used in 2.3, and compute the fitting factor between these signals and the aligned spin template bank. The results of this are shown in FIG.10. To decrease the computational cost of this test, we only calculated the overlaps between a signal and templates that were within a range of $\pm 0.1 M_{\odot}$ in chirp mass. This is reasonable because the overlap will decrease rapidly with small changes in chirp mass, therefore we expect templates with very different values of chirp mass to have low overlaps with each other. We verified that this approach did not cause us to underestimate the fitting-factor of our banks.

We can now compare the results obtained in this section, using our aligned-spin template bank, with the results obtained in section 2.3, using a non-spinning template

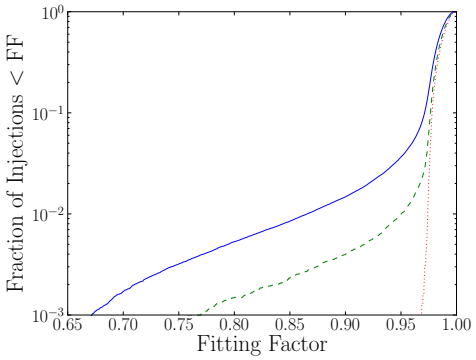

Figure 10: The distribution of fitting factors obtained by searching for the precessing signals described in section 2.3 with component spins up to 0.4 (blue solid line), 0.2 (green dashed line), and 0.05 (red dotted line) using the aligned spin BNS template bank described in section 2.4 and the advanced LIGO, zero-detuned, high-power PSD with a 15Hz lower frequency cutoff.

bank. One can clearly see an improvement in the distribution of fitting factors when using the aligned spin template bank. The fraction of signals that fall below a fitting factor of 0.97, when the spin magnitudes are restricted to 0.4, falls from 59% to 9%. We also see an improvement for signals that have spin magnitudes restricted to 0.05, where the fraction of signals falling below a fitting factor of 0.97 drops from 6% to 0.2%. We can also compare the performance of the aligned-spin bank to that of the non-spinning bank as a function of the maximum spin magnitude, as shown in Figure 11. From this Figure we can see that regardless of the maximum component spin, the aligned spin bank will greatly reduce the number of signals recovered with fitting factors less than 0.97.

A small fraction of signals fall below a FF of 0.97, even when using the new aligned-spin template bank. We expect that these poor matches with the aligned template bank are due to precession. In general, precessional effects will not be important in BNS systems as the orbital angular momentum is significantly larger than the component spins. In such cases there is only a small angle between the total and orbital angular momenta and precession has only a small effect on the waveform.

However, there is a small region of parameter space where precessional effects *will* have an effect for BNS systems. Using the model of Ref. [**?**], applied to the small precession angles in BNS systems, we can predict for which systems precession will be most important. The orientation of a precessing binary must be defined using

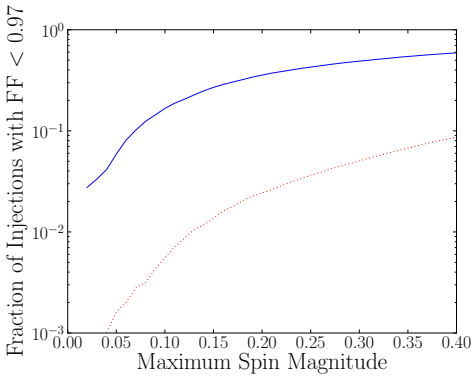

Figure 11: The fraction of the precessing signals described in section 2.3 recovered with a fitting factor less than 0.97 as a function of the maximum component spin. Shown for the non-spinning BNS template bank described in section 2.3 (blue solid line), and the aligned spin BNS template bank described in section 2.4 (red dotted line). The advanced LIGO, zero-detuned, high-power PSD with a 15Hz lower frequency cutoff was used when computing the fitting factors.

the total angular momentum rather than the orbital angular momentum as done with non-precessing binaries. The orientations with the worst matches should be those where the system is edge-on (angular momentum perpendicular to the viewing direction) and where the detector is nearly insensitive to the plus polarization and only sees the cross polarization (a binary overhead of the detector would have its angular momentum oriented 45° between the arms of the detector). We find that this is indeed the case; in fact, all cases with fitting factors less than 0.95 are close to this configuration. All of these cases also have biases in the recovered mass and spin parameters due to the secular effects of precession on the phasing of the waveform.

2.7 Conclusion

In this work we have investigated the effects of neglecting spin when searching for binary neutron star systems in aLIGO and AdV. We have found that, if component spins in binary neutron star systems are as large as 0.4, then neutron star spin cannot be neglected, and there is a non-trivial loss in signal-to-noise ratio even if the maximum spin is restricted to be less than 0.05. We have developed a new algorithm for placing an aligned spin template bank in the BNS parameter space. We have shown that this bank works for aligned spin systems and have demonstrated that

it does significantly better for generic, precessing BNS systems than the traditional non-spinning bank. However, for the BNS aligned spin $\chi_i < 0.4$ parameter space the aligned spin bank requires approximately five times as many templates as the non-spinning bank. This increased number of templates will increase the computational cost of the search and increase the number of background events, so needs to be balanced against the potential gain in being able to cover a larger region of parameter space. A further advantage of our method is the ease with which it can be incorporated into existing or future search pipelines, which include the use of signal-based vetoes [**?**] and coincidence algorithms [**?**]. In future work we will investigate how this template bank performs in data from the aLIGO and AdV detectors which includes non-Gaussian and non-stationary noise features. Finally we note that the method proposed in this work should be applicable wherever the TaylorF2 waveforms closely represent actual gravitational waveforms. In a future work we will investigate how well this method performs in the binary black hole and neutron-star, black-hole regions of the parameter space. Wherever the TaylorF2 approximation begins to break down, a stochastic bank placement may still be the most viable option.

Acknowledgements

The authors are greatful to Stefan Ballmer, Stephen Fairhurst, Eliu Huerta, Drew Keppel, Prayush Kumar, Frank Ohme, Ben Owen, Reinhard Prix, Peter Saulson, B.S. Sathyaprakash, John Veitch, Matthew West and Karl Wette for helpful discussions. DB, IH and AN are supported by NSF award PHY-0847611. IH and AN are also also supported by NSF award PHY-0854812. AL was supported by NSF grant PHY-0855589 and the Max Planck Gesellschaft. DB and IH are also supported by a Cottrell Scholar award from the Research Corporation for Science Advancement. Computations used in this work were performed on the Syracuse University Gravitation and Relativity cluster, which is supported by NSF awards PHY-1040231, PHY-0600953 and PHY-1104371.

Chapter 3

NSBH Accuracy

3.1 abstract

Gravitational waves radiated by the coalescence of compact-object binaries containing a neutron star and a black hole are one of the most interesting sources for the ground-based gravitational-wave observatories Advanced LIGO and Advanced Virgo. Advanced LIGO will be sensitive to the inspiral of a $1.4\,M_\odot$ neutron star into a $10\,M_\odot$ black hole to a maximum distance of ~ 900 Mpc. Achieving this sensitivity and extracting the physics imprinted in observed signals requires accurate modeling of the binary to construct template waveforms. In a neutron star–black hole binary, the black hole may have significant angular momentum (spin), which affects the phase evolution of the emitted gravitational waves. We investigate the ability of currently available post-Newtonian templates to model the gravitational waves emitted during the inspiral phase of neutron star–black hole binaries. We restrict to the case where the spin of the black hole is aligned with the orbital angular momentum and compare several post-Newtonian approximants. We examine restricted amplitude post-Newtonian waveforms that are accurate to third-and-a-half post-Newtonian order in the orbital dynamics and complete to second-and-a-half post-Newtonian order in the spin dynamics. We also consider post-Newtonian waveforms that include the recently derived third-and-a-half post-Newtonian order spin-orbit correction and the third post-Newtonian order spin-orbit tail correction. We compare these post-Newtonian approximants to the effective-one-body waveforms for spin-aligned binaries. For all of these waveform families, we find that there is a large disagreement between different

waveform approximants starting at low to moderate black hole spins, particularly for binaries where the spin is anti-aligned with the orbital angular momentum. The match between the TaylorT4 and TaylorF2 approximants is ~ 0.8 for a binary with $m_{BH}/m_{NS} \sim 4$ and $\chi_{BH} = cJ_{BH}/Gm_{BH}^2 \sim 0.4$. We show that the divergence between the gravitational waveforms begins in the early inspiral at $v \sim 0.2$ for $\chi_{BH} \sim 0.4$. Post-Newtonian spin corrections beyond those currently known will be required for optimal detection searches and to measure the parameters of neutron star–black hole binaries. The strong dependence of the gravitational-wave signal on the spin dynamics will make it possible to extract significant astrophysical information from detected systems with Advanced LIGO and Advanced Virgo.

3.2 Introduction

Compact object binaries are likely to be the first source detected by the The Advanced Laser Interferometer Gravitational Wave Observatory (aLIGO) [**?**] and Advanced Virgo (AdV) [**?**]. These detectors will be sensitive to the gravitational waves radiated as the orbital frequency of the binary sweeps upwards from $\sim$ 5–10 Hz to the point at which the compact objects coalesce [**?**]. Binaries containing a neutron-star–black-hole (NSBH) have a predicted coalescence rate of 0.2–300 yr^{-1} within the sensitive volume of aLIGO [**?**], making them an important source for these observatories. The observation of a NSBH by aLIGO would be the first conclusive detection of this class of compact-object binary. Gravitational-wave observations of NSBH binaries will allow us to explore the central engine of short, hard gamma-ray bursts, shed light on models of stellar evolution and core collapse, and investigate the dynamics of compact objects in the strong-field regime [**?**, **?**, **?**, **?**, **?**, **?**, **?**]. Achieving aLIGO's optimal sensitivity to NSBH binaries and exploring their physics requires accurate modeling of the gravitational waves emitted over many hundreds of orbits as the signal sweeps through the detector's sensitive band. For binary neutron star (BNS) systems the mass ratio between the two neutron stars is small and the angular momenta of the neutron stars (the neutron stars' spins) is low. In this case, the emitted waves are well modeled by Post-Newtonian (PN) theory [**?**, **?**, **?**]. However, NSBH binaries can have significantly larger mass ratios and the spin of the black hole can be much larger than that of a neutron star. The combined effects of mass ratio and spin present challenges

in constructing accurate gravitational waveform models for NSBH systems, compared to BNS systems. In this paper we investigate how accurately current theoretical models simulate NSBH gravitational waveforms within the sensitive frequency band of aLIGO.

Although no NSBH binaries have been directly observed, both black holes (BHs) and neutron stars (NSs) have been observed in other binary systems. Several BNS systems and neutron star-white dwarf (NSWD) systems have been observed by detecting their electromagnetic signatures. Electromagnetic observations suggest that the NS mass distribution in BNS peaks at $1.35M_\odot$–$1.5M_\odot$ with a narrow width [**?**], although NSs in globular clusters seem to have a considerably wider mass distribution [**?**]. There is also evidence that a neutron star in one system has a mass as high as $\sim 3M_\odot$ [**?**]. The dimensionless spin magnitude $\chi = cJ/Gm^2$ for NSs is constrained by possible NS equations of state to a maximum of 0.7 [**?**]. The fastest observed pulsar has a spin period of 1.4 ms [**?**], corresponding to a $\chi \sim 0.4$, and the most rapidly spinning observed NS in a binary, J0737–3039A, has a spin of only $\chi \sim 0.05$. The observational data for BHs is more limited than for NSs. Studies of BHs in low-mass X-ray binaries suggest a mass distribution of $7.8 \pm 1.2M_\odot$ [**?**]. This extends to $8-11 \pm 2-4M_\odot$ when 5 high-mass, wind-fed, X-ray binary systems are included [**?**]. For BHs there is evidence for a broad distribution of spin magnitudes [**?**], although general relativity limits it to be $\chi < 1$. Given the uncertainties in the masses and spins of NSBH binaries, we consider a fairly broad mass and spin distribution when investigating the accuracy of NSBH waveforms. In this paper, we consider NSBH binaries with the NS mass between 1 and $3\,M_\odot$, the BH mass between 3 and $15\,M_\odot$, the NS spin between 0 and 0.05 and the BH spin between 0 and 1. Between these limits, the distributions of mass and spin are all assumed to be uniform.

Gravitational-wave detectors are sensitive to the phase evolution of the waves radiated by the binary. PN theory can be used to compute the energy of a compact binary $E(v)$ and the flux radiated in gravitational waves $\mathcal{F}(v)$ in terms of the invariant velocity $v = (\pi M f)^{1/3}$, where $M = m_1 + m_2$ is the total mass of the binary, and f is the gravitational-wave frequency [**?**]. By solving the energy balance equation $dE/dt = -\mathcal{F}$, we can obtain expressions for the gravitational-wave phase as a function of time $\phi(t)$ or, equivalently, the Fourier phase of the waves as a function of frequency $\Psi(f)$. At leading order, the gravitational wave phase depends only on the chirp mass

$\mathcal{M}_c = (m_1 m_2)^{3/5}/(m_1+m_2)^{1/5}$ [?]. Beyond leading order, the waveforms also depend on the symmetric mass ratio $\eta = m_1 m_2/(m_1+m_2)^2$ [?, ?, ?, ?, ?, ?], with spin-orbit corrections entering at the third correction beyond leading order [?, ?, ?, ?, ?].

There are several different ways in which to solve the energy balance equation to obtain the gravitational-wave phase measurable by aLIGO; these different methods are known as PN *approximants.* While the convergence of the full PN series is not guaranteed, for BNS systems in Advanced LIGO, the available PN approximants produce waveforms that are indistinguishable for a given binary and are reliable for use in detection searches and parameter measurement [?, ?, ?]. However, for NSBH binaries the total mass, and hence the PN expansion parameter v, is larger. The mass ratio and spin corrections are also more significant. In this paper, we investigate the accuracy of waveforms generated by different PN approximants for observing NSBH binaries with aLIGO. To do this, one could compare subsequent terms in the PN expansion and determine the effect of neglecting them. However, in the case of systems whose component objects are spinning, only terms up to 2.5PN order are completely known [?, ?, ?]. This represents the leading order (1.5PN) and next-to-leading order (2.5PN) spin-orbit, along with the leading order (2.0PN) spin-spin contributions to the phasing [?, ?, ?]. We choose to compare approximants that are constructed with terms up to the same PN order, but that use inversely related differential equations to solve for the orbital dynamics, in addition to comparing to approximants that include higher order spin-related corrections at partially derived orders [?, ?]. These methods both have the effect of testing how well the PN series has converged. We also present a comparison between waveforms from these PN approximants where we fix the mass and spin parameters of the objects in order to understand when in the inspiral the waveforms diverge.

We consider two families of PN approximants for binaries where the spin of the black hole is aligned with the orbital angular momentum: TaylorT2 [?, ?, ?] and TaylorT4 [?]. In these models, we include all the completely known orbital evolution terms (up to 3.5PN order) [?, ?, ?, ?, ?, ?] and all the completely known spin-related terms (up to 2.5PN order) [?, ?, ?, ?, ?]. Restricting to systems where the spin angular momenta are aligned (or anti-aligned) with the orbital angular momentum means that the plane of the binary does not precess, simplifying our comparisons. However, this study captures the dominant effect of spin on the waveforms [?]. In a

separate paper, we investigate the effect of precession on detection searches [**?**]. We also consider the effective-one-body model as described in Ref. [**?**]. We restrict to comparing the inspiral portion of approximants. Even at the upper range of masses we consider, $(3 + 15)M_{\odot}$, it has been shown in the case of numerically modelled binary black hole waveforms that inspiral-only template banks recover $> 95\%$ of the signal power [**?**, **?**]. We separately consider models that include spin-related terms up to 3.5PN order [**?**, **?**]. Spin-orbit tail (3.0PN) and next-to-next-to-leading order spin-orbit (3.5PN) contributions to the phasing are known. However, these orders are incomplete as there are also unknown spin corrections at 3.0PN and 3.5PN, including spin-spin and (spin-induced) octupole-monopole couplings.

In Fig. 12 we show the distance an optimally oriented system would be observed at signal-to-noise ratio (SNR) 8 (the horizon distance), for a $1.4M_{\odot} - 10M_{\odot}$ NSBH system, as a function of the spin of the black hole, for both the aLIGO zero-detuned, high-power sensitivity curve and a plausible range of early aLIGO sensitivities [**?**]. Systems where the spin of the black hole is large in magnitude and aligned with the orbital angular momentum can be seen from a greater distance than systems where the spin is small or anti-aligned. Achieving this sensitivity requires NSBH waveforms that do not incur a significant loss in SNR when used as search templates [**?**]. Furthermore, extracting the physics from observed signals requires faithful templates for parameter measurement.

We find that no presently available waveform model is sufficiently accurate for use in aLIGO NSBH searches or parameter measurement. Our key results, Figs. 13-17, show the match between the various waveform families considered here. There is a significant disagreement between the PN approximants we have examined, even at at low ($\chi \sim 0.4$) spins and small ($m_{BH}/m_{NS} \sim 4$) mass ratios for TaylorF2 and TaylorT4. The match decreases as these increase with matches as low as ~ 0.1 observed. This motivates the need to compute higher order PN spin corrections.

Our present knowledge of NSBH waveforms will limit the ability of gravitational-wave observatories to accurately determine source parameters from the detected signals and may hinder their detection. Further analytical and numerical modeling of NSBH systems will be needed before aLIGO comes online in 2015 and reaches full sensitivity in $\sim$ 2019 [**?**].

The remainder of this paper is organized as follows. In Sec. 3.3, we describe

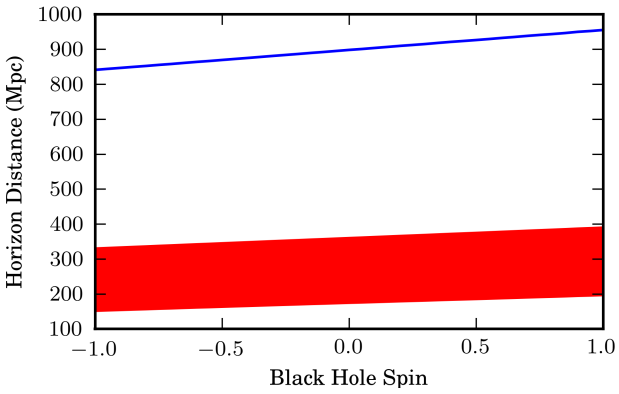

Figure 12: The horizon distance as a function of the spin of the black hole for a $1.4M_\odot - 10M_\odot$ NSBH system, for both the aLIGO zero-detuned, high-power aLIGO sensitivity curve (blue) and plausible early aLIGO detector sensitivities (red), with a 15 Hz lower frequency cutoff. Results are obtained using the TaylorT4 approximant including only the complete spin terms up to 2.5PN. Note that aLIGO will be sensitive to NSBH systems out to ~ 900 Mpc, and there will be increased sensitivity for systems with aligned black hole spins with large magnitudes.

the construction of the PN approximants used and Sec. 3.4 describes our method of comparing them. In Sec. 3.5 we show the results of comparing different PN approximants, and show that there is a large discrepancy between the waveforms for NSBH binaries at relatively low black hole spins. In Sec. 3.6 we construct a new frequency domain approximant that is designed to agree with TaylorT4. This is followed by a comparison of the time domain approximants to their frequency domain counterparts in Sec. 3.7, where we demonstrate that they largely agree. Finally, in Sec. 3.8 and Sec. 3.9 we investigate where in the inspiral the disagreement between the waveform families becomes important. We demonstrate that the divergence occurs at surprisingly low velocities for even modest black hole spins. Finally in Sec. 3.10 we investigate whether maximizing over the mass and spin parameters of the waveform can improve present models, and investigate the accuracy of the waveforms for early aLIGO observations when the detectors will have reduced low-frequency sensitivity when compared to the ultimate sensitivity.

3.3 Constructing post-Newtonian Waveforms

We examine the accuracy and convergence of currently known waveforms for NSBH binaries by comparing approximants constructed using the PN approximations of the binary's equation of motion and gravitational radiation. To obtain the gravitational-wave phase from these quantities, we assume that the binary evolves adiabatically through a series of quasi-circular orbits. This is a reasonable approximation as gravitational radiation is expected to circularize the orbits of isolated binaries [**?**]. In this limit, the equations of motion reduce to series expansions of the center-of-mass energy $E(v)$ and gravitational-wave flux $\mathcal{F}(v)$, which are expanded as a power series in the orbital velocity v around $v = 0$. They are given as

$$E(v) = E_{\rm N} v^2 \left(1 + \sum_{n=2}^{6} E_i v^i\right), \tag{3.1}$$

$$F(v) = F_{\rm N} v^{10} \left(1 + \sum_{n=2}^{7} F_i v^i\right), \tag{3.2}$$

where the coefficients $\{E_{\rm N}, E_i, F_{\rm N}, F_i\}$ are defined in Appendix 3.12. For terms not involving the spin of the objects, the energy is known to order v^6, while the flux is known to v^7, referred to as 3.0PN and 3.5PN, respectively. At order 3.0PN, the flux contains terms proportional to both v^6 and $v^6 \log v$; which are regarded to be of the same order. Complete terms involving the spins of the objects first appear as spin-orbit couplings at 1.5PN order, with spin-spin couplings entering at 2PN order, and next-to-leading order spin-orbit couplings known at 2.5PN order.

We use the assumption that these systems are evolving independently to relate the PN energy and gravitational-wave flux equations, i.e. the loss of energy of the system is given by the gravitational-wave flux

$$\frac{dE}{dt} = -\mathcal{F}. \tag{3.3}$$

This can be re-arranged to give an expression for the time evolution of the orbital velocity,

$$\frac{dv}{dt} = -\frac{\mathcal{F}(v)}{E'(v)}, \tag{3.4}$$

where $E'(v) = dE/dv$. The orbital evolution can be transformed to the gravitational waveform by matching the near-zone gravitational potentials to the wave zone. The amplitude of gravitational waves approximated in this way are given by the PN expansion of the amplitude. This gives different amplitudes for different modes of the orbital frequency. The dominant gravitational-wave frequency f is given by twice the orbital frequency, which is related to the orbital velocity by $v = (\pi M f)^{1/3}$. The orbital phase is therefore given by

$$\frac{d\phi}{dt} = \frac{v^3}{M}, \tag{3.5}$$

and the phase of the dominant gravitational-wave mode is twice the orbital phase. Here, we will only expand the gravitational-wave amplitude to Newtonian order (0PN), which, when combined with the phase, is referred to as a restricted PN waveform.

Solutions $v(t)$ and $\phi(t)$ to Eqs. (3.4) and (3.5) can be used to construct the plus and cross polarizations and the observed gravitational waveform. For restricted waveforms, these are:

$$h_+(t) = -\frac{2\,M\,\eta}{D_L}\,v^2\,(1+\cos^2\theta)\,\cos 2\phi(t)\ , \tag{3.6}$$

$$h_\times(t) = -\frac{2\,M\,\eta}{D_L}\,v^2\,2\cos\theta\,\sin 2\phi(t)\ , \tag{3.7}$$

$$h(t) = F_+\,h_+(t) + F_\times\,h_\times(t)\ . \tag{3.8}$$

Here F_+ and $F_\times$ are the antenna pattern functions of the detector, D_L is the luminosity distance between the binary and observer, and θ is the inclination angle between the orbital angular momentum of the binary and the direction of gravitational-wave propagation: $\cos\theta = \hat{L}\cdot\hat{N}$. Thus, a non-precessing, restricted PN waveform is fully specified by $v(t)$ and $\phi(t)$ (or equivalently $t(v)$ and $\phi(v)$).

We now have the ingredients necessary to produce the TaylorT2 and TaylorT4 families of approximants, which we describe in the following sections.

3.3.1 TaylorT4

The TaylorT4 approximant, introduced in [**?**], is formed by numerically solving the differential equation

$$\frac{dv}{dt} = \left[\frac{-\mathcal{F}(v)}{E'(v)}\right]_k = A_k(v). \tag{3.9}$$

The notation $[Q]_k$ indicates that the quantity Q is to be truncated at v^k order. Terms containing pieces logarithmic in v are considered to contribute at the order given by the non-logarithmic part. Thus waveforms expanded to 3.5PN order in the phase would be truncated at $k = 7$. We use A_k as shorthand for the truncated quantity that is used as the expression for dv/dt.

The resulting differential equation, given explicitly in Appendix 3.13.1, is non-linear and therefore must be solved numerically. The result is a function $v(t)$. The phase can then be calculated by

$$\frac{d\phi}{dt} = \frac{v(t)^3}{M}. \tag{3.10}$$

The phase is integrated from a fiducial starting frequency up to the minimum energy condition (MECO), which is defined by

$$\frac{dE(v)}{dv} = 0. \tag{3.11}$$

The MECO frequency is where we consider the adiabatic approximation to have broken down. Note that the MECO frequency is dependent on not only the masses but also the spins of the objects; specifically, systems where the objects' spins are aligned with the orbital angular momentum will have a higher MECO frequecy. When the partial spin-related terms at 3.0PN and 3.5PN are included, however, there are regions of the NSBH parameter space for which the MECO condition is never satisfied. For these cases, we impose that the rate of increase in frequency must not decrease (i.e. we stop if $dv/dt \leq 0$), and that the characteristic velocity of the binary is less than c (i.e. we stop if $v \geq 1$). We terminate the waveforms as soon as any of these stopping conditions are met.

3.3.2 TaylorT2

In contrast to the TaylorT4 approximant, the TaylorT2 approximant is constructed by expanding t in terms of v and truncating the expression to consistent PN order. We first construct the quantity

$$\frac{dt}{dv} = \left[\frac{E'(v)}{-\mathcal{F}(v)}\right]_k = B_k(v). \tag{3.12}$$

This can be combined with the integral of (3.5) and solved in closed form as a perturbative expansion in v,

$$\phi(v) = \int \frac{v^3}{M} B_k(v) dv. \tag{3.13}$$

The explicit result of this integral is given in Appendix 3.13.2. Similar to TaylorT4, the phase is generally calculated up to the MECO frequency. However, for some points of parameter space, this formulation can result in a frequency that is not monotonic below the MECO frequency. As with TaylorT4, we stop the waveform evolution with $dv/dt \leq 0$ or $v \geq 1$.

A related approximant can be computed directly in the frequency domain by using the stationary phase approximation [**?**, **?**]. This approximant is called TaylorF2 and can be expressed as an analytic expression of the form

$$\phi(f) = A(f) e^{i\psi(f)}, \tag{3.14}$$

where the phase takes the form

$$\psi(f) = \sum_{i=0}^{7} \sum_{j=0}^{1} \lambda_{i,j} f^{(i-5)/3} \log^j f. \tag{3.15}$$

The full expressions for the amplitude and phase are given in Appendix 3.13.3. Because the stationary phase approximation is generally valid, the TaylorT2 and TaylorF2 approximants are nearly indistinguishable [**?**]. An advantage of the TaylorF2 approximant comes from the fact that it can be analytically calculated in the frequency domain. In practice, waveforms that are generated in the frequency domain without the use of integration are less computationally costly, and so searches for gravitational waves from inspiraling binary systems have been performed using the

TaylorF2 approximant [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?].

3.3.3 SEOBNRv1

An additional approximant we use is the spinning effective one-body model (SEOBNRv1), presented in Ref. [?]. This approximant incorporates the results of black hole perturbation theory, the self-force formalism and PN results. The model has been calibrated to numerical relativity simulations, including simulations where the objects' spins were (anti-) aligned with the orbital angular momentum and had magnitudes of $\chi \pm 0.4$. In order to compare these waveforms more fairly with the PN approximants that only model the inspiral, we truncate this model before the merger section of the waveform.

The implemented versions of SEOBNRv1 are currently limited to $\chi \leq 0.6$. To further extend the model would require better modeling of the plunge physics and possibly the computation and incorporation of additional PN terms.

3.4 Computing faithfulness

Searches for gravitational waves from compact binary coalescences utilize matched-filtering [?, ?], in which the signal model is correlated with the detector output to construct a signal-to-noise ratio. If the signal model does not accurately capture the true gravitational waveform, then the signal-to-noise ratio, and hence the distance to which the detector can see signals at a given false alarm rate, will decrease. Matched-filtering therefore relies on the accuracy of the models. We quantify the agreement between waveform families by computing the match, or *faithfulness* of the waveforms, defined as follows. We define the noise-weighted inner product between two gravitational waveforms, h_1 and h_2, to be

$$(h_1|h_2) = 4\Re \int_0^\infty \frac{\tilde{h}_1(f)\tilde{h}_2^*(f)}{S_n(f)} df, \tag{3.16}$$

where

$$\tilde{h}_1(f) = \int_0^\infty h_1(t) e^{-2\pi i f t}\, dt \tag{3.17}$$

is the Fourier transform of $h_1(t)$, and $S_n(f)$ denotes the one-sided power spectral density of the gravitational-wave detector's noise. In practice, the signals are discretely sampled so the upper frequency limit is the Nyquist frequency of the data, and the lower frequency limit of the integral is set by the detector's low-frequency sensitivity [**?**]. We define the normalized overlap between two waveforms h_1 and h_2 as

$$(h_1|h_2) = \frac{(h_1|h_2)}{\sqrt{(h_1|h_1)(h_2|h_2)}}. \tag{3.18}$$

The match between two waveforms is obtained by maximizing the overlap over the phase of the waveform and ϕ_c and any time shifts t_c between h_1 and h_2

$$\mathcal{M}(h_1, h_2) = \max_{\phi_c, t_c}(h_1|h_2(\phi_c, t_c)), \tag{3.19}$$

where the shifted waveform can be constructed as

$$\tilde{h}(\phi_c, t_c) = \tilde{h}e^{i(\phi_c - 2\pi f t_c)}. \tag{3.20}$$

The *faithfulness* of representing a waveform from a given PN family with that of another is described by the match between the two waveforms when the same physical parameters are used as input to the models. As both models describe the same physical source, the match should be unity. Any deviation is due to the variation between models and the match gives the fractional loss in signal-to-noise ratio that will result.

3.5 Post-Newtonian approximant faithfulness comparison

In this section we compare the faithfulness between waveforms from different PN approximants where we choose the physical parameters to be consistent with NSBH sources. We also consider how the waveforms from the PN approximants compare to the waveforms from the SEOBNRv1 effective-one-body model [**?**]. Lastly, we consider the effect of including the spin-related terms at only partially derived orders. We model the sensitivity of second generation gravitational-wave detectors with the aLIGO zero-detuned, high-power sensitivity curve [**?**]. For this study we use a lower

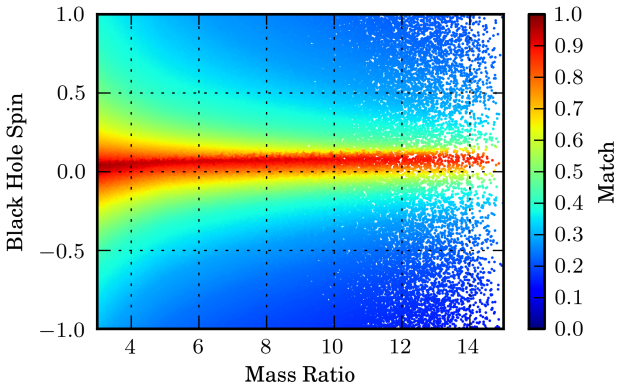

Figure 13: The match between the TaylorF2 and TaylorT4 approximants as a function of the spin of the black hole and the mass ratio of the system. Only the completely known spin-related corrections up to 2.5PN are included. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve and a 15Hz lower frequency cutoff. A significant reduction in match is seen for even moderate spins $\chi \sim 0.3$ and low mass ratios $m_{bh}/m_{ns} \sim 4$. The approximants also begin to disagree for non-spinning systems as the mass ratio increases.

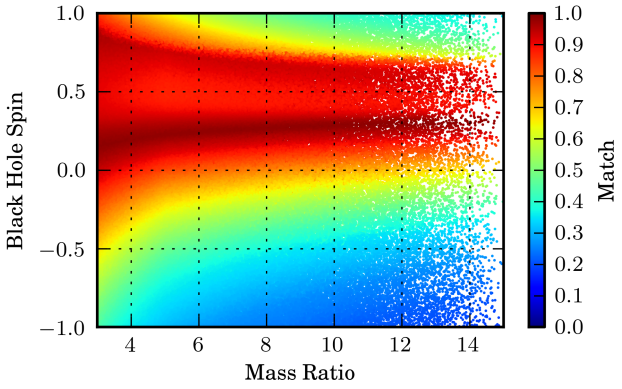

Figure 14: The match between the TaylorF2 and TaylorT4 approximants as a function of black hole spin and mass ratio. Both models include the next-to-next-to-leading spin-orbit (3.5PN) and spin-orbit tail terms (3.0PN). In comparison to Fig. 13, the additional terms have improved the agreement for moderately spinning aligned spin systems, however, the match is still ~ 0.8 for $\chi \sim 0.5$ at all mass ratios.

frequency cutoff of 15Hz since it is not expected that detectors will have significant sensitivity below this frequency. We consider the effect of increasing this low-frequency cutoff to simulate early aLIGO sensitivities in Sec. 3.10.

In Fig. 13, we examine the faithfulness of NSBH waveforms by computing the match between the TaylorF2 and TaylorT4 PN approximants. The TaylorT4 approximant was used to simulate NSBH binaries in LIGO's previous gravitational-wave searches, and the TaylorF2 family is used as the templates for detection [**?**]. In order to focus on the mismatches primarily due to phase differences between the models, the frequency cutoff of the TaylorF2 waveform is made to agree with the ending frequency of the TaylorT4 waveform. We see that the agreement between the two models is primarily influenced by the magnitude of the black hole's spin, and secondarily by the mass ratio. There is a noticeable drop in match at higher mass ratios, even when the spin of the black hole is zero. As expected, the best agreement is seen when the black hole's spin is small and the black hole and neutron star have comparable masses. However, this plot shows that there is a *substantial* disagreement between these approximants for even moderately low black hole spins ($\chi \sim 0.3$), which increases as the spin of the black hole increases. We note that the effect on the match due to the spin of the neutron star is negligible in all areas. In Fig. 14 we compare the TaylorF2 and TaylorT4 models, with the inclusion of the spin-orbit tail (3.0PN) and next-to-next-to-leading spin-orbit (3.5PN) corrections recently computed in Refs. [**?**, **?**]. In comparison to Fig. 13, the agreement is improved for aligned spins with moderate magnitudes. However, these approximants maintain a poor level of overall agreement, with matches of only ~ 0.8 at $\chi \sim 0.5$ for all mass ratios, and even lower matches for anti-aligned systems. Figs. 15 and 16 compare the TaylorT2 and TaylorT4 approximants with and without these additional spin terms. We see that TaylorT4 is especially sensitive to the additional corrections. In both cases, however, we note that the additional terms have caused a significant change in the waveforms, as indicated by the low matches, demonstrating that the expansion has not yet sufficiently converged to produce reliable waveforms for parameter estimation.

In Fig. 17 we compare the SEOBNRv1 model to the PN models TaylorF2 and TaylorT4. Since the SEOBNRv1 model is not valid for large values of χ [**?**] we restrict $\chi < 0.6$ and only report matches below this limit. We see that, similar to the comparison between TaylorF2 and TaylorT4, these models also have large

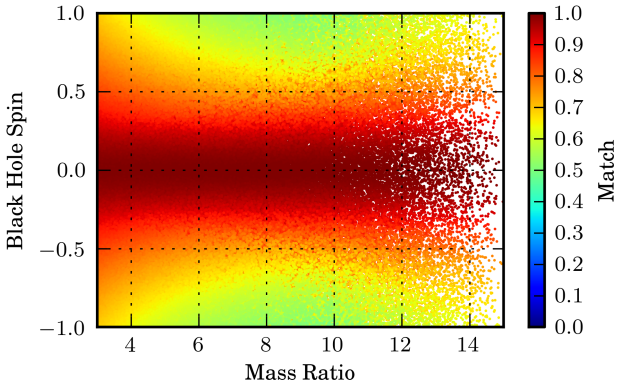

Figure 15: The match between TaylorF2 with 2.5PN spin corrections and TaylorF2 including the next-to-next-to-leading spin-orbit (3.5PN) and spin-orbit tail terms (3.0PN), as a function of the spin of the black hole and the mass ratio of the system. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve and a 15Hz lower frequency cutoff. Although there is agreement where the spins are low $\chi < 0.2$, the match quickly drops as the spin of the black hole increases, so that the match is already ~ 0.7 for $\chi \sim 0.5$.

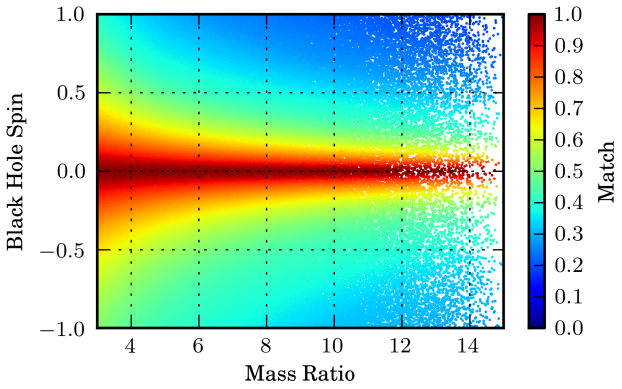

Figure 16: The match between TaylorT4 with 2.5PN spin corrections and TaylorT4 including the next-to-next-to-leading spin-orbit (3.5PN) and spin-orbit tail terms (3.0PN), as a function of the spin of the black hole and the mass ratio of the system. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve and a 15Hz lower frequency cutoff. In comparison to Fig. 15, the approximant is more noticeably changed by the additional terms. For a mass ratio of 8, the match has already fallen to ~ 0.7 for $\chi \sim 0.15$.

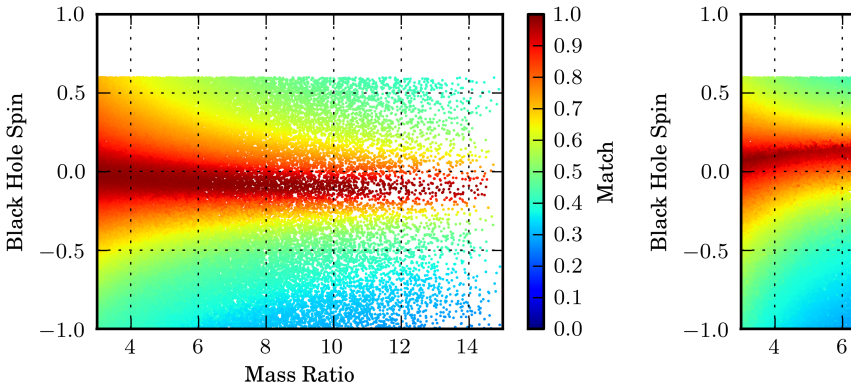

Figure 17: The match between the TaylorF2 (left) or TaylorT4 (right) and SEOBNRv1 approxim
2.5PN. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve wit
reduction in match where spin of the black hole is only moderate. Note, however, that the PN app
with each other.

mismatches when the spin of the black hole is nonzero. The large discrepancy between the waveform families indicates that higher order PN correction terms are required. This may also pose significant problems for parameter estimation of NSBH sources.

3.6 The TaylorR2F4 approximant

In the previous section, we found a surprisingly large disagreement between the TaylorF2 and TaylorT4 PN approximants when compared with waveform parameters appropriate for NSBH systems. We would like to distinguish how much of this is due to differences between time domain and frequency domain approximants, and how much of this is due to differences between the formulation of the two PN families. This can easily be performed for the TaylorF2 and TaylorT2 approximants, however we need to construct an equivalent frequency domain version of TaylorT4 to complete the comparison.

By analogy with TaylorF1 and TaylorF2 [**?**, **?**], TaylorF4 is obtained by numerically integrating the reciprocal of Eq. (4.11) in the frequency domain,

$$dt/dv = 1/A_k(v). \tag{3.21}$$

However, this does not elucidate the differences between the TaylorT4 and TaylorF2 approximants. Instead, we construct an analytical approximation to the TaylorF4

approximant, which we call TaylorR2F4, by expanding Eq. (3.21) in powers of v. In order to make this series finite, we truncate these additional terms at an order in v higher than the order where the PN expansion of the energy and flux were truncated,

$$\frac{dt}{dv} = \left[\frac{1}{A_k(v)}\right]_l = B_k(v) + R_{kl}(v) = C_{kl}(v). \tag{3.22}$$

Here $B_k(v)$ is the same as in the TaylorT2 approximant and $R_{kl}(v)$ are the terms from order v^{k+1} up to order v^l. It is important to note that this produces a power series that is identical to the TaylorF2 approximant up to the point where (4.5) was truncated. Thus, terms of higher order in v account for the differences between the TaylorT2 and TaylorT4 approximants.

In sec. 3.7 we show that TaylorR2F4 agrees well with the TaylorT4 approximant when expanded to v^9 or v^{12}, which we shall see in the next section. As noted above, the second expansion in the TaylorR2F4 approximant is a different expansion than the PN expansion of the energy and flux. The Fourier phase for the TaylorR2F4 approximant can be obtained from (4.6) where $B_k(v)$ is replaced by $C_{kl}(v)$. This is given up to order v^N as

$$\psi_{\mathrm{R2F4}}(f) = \psi_{F2}(f) + \sum_{i=6}^{N}\sum_{j=0}^{N} \lambda_{i,j} f^{(i-5)/3} \log^j f, \tag{3.23}$$

where the form of these expressions up to $N = 12$ can be found in Appendix 3.13.4. Because this approximant can be analytically expressed in the frequency domain, it can be generated relatively cheaply compared to TaylorT4. This means that it has the potential to be used where computational efficiency and a higher degree of agreement with TaylorT4 is desired. We note that the frequency-domain approximants are much faster than their time-domain counterparts, which must integrate differential equations and perform a Fourier transform. Therefore, they are especially useful in computational problems which are waveform-generation limited, such as parameter estimation of signals [**?**].

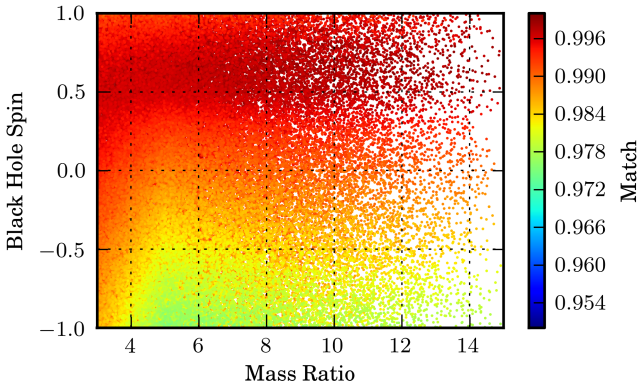

Figure 18: The match between TaylorF2 and TaylorT2. Both include spin corrections up to 2.5PN order. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve and a 15Hz lower frequency cutoff. We see that the F2 and T2 approximants largely agree. The discrepancy between the two approixmants can be reduced by expanding the frequency sweep of the TaylorF2 approximant's amplitude to higher PN orders. However, there is different Gibbs phenomena between the two approximants that will cause a discrepancy.

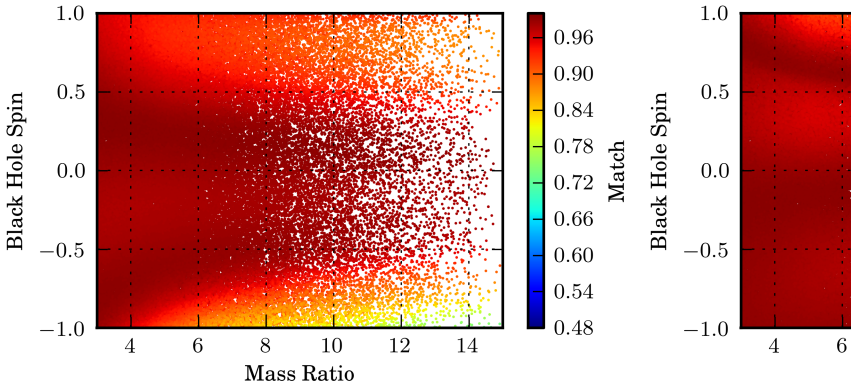

Figure 19: The match between TaylorT4 and TaylorR2F4. Both models include spin corrections up
(right). Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve an
over a broad range of parameters, with some visible exceptions. Expanding up to orde

3.7 Comparison of Frequency to Time Domain Approximants

In this section, we investigate to what extent the discrepancy between the waveform families that was demonstrated in Sec. 3.5 is due to the difference between expressing approximants in the frequency and time domain alone. We compare the new TaylorR2F4 approximant from Sec. 3.6, and TaylorF2, to their time domain equivalents.

We find that TaylorF2 waveforms are a good representation of TaylorT2 waveforms, even when we consider waveforms from NSBH systems where the component objects are spinning. This can be seen in Figure 18, which shows the match between the TaylorF2 and TaylorT2 models. In that figure, the ending frequency of both models is made to be the same, which is accomplished by terminating the TaylorF2 waveforms at the frequency where the generation of the equivalent TaylorT2 waveforms terminated. We find that the TaylorF2 and TaylorT2 waveforms agree to better than $\gtrsim 95.7\%$ for the entire region investigated. For systems where the black hole spin was positively aligned with the orbital angular momentum, the match is $\gtrsim 97.9\%$. The discrepancy between these two models is in part due to expanding to only Newtonian order the frequency sweep associated with the stationary phase approximation of the TaylorF2 approximant. In addition, part of the discrepancy results from Gibbs phenomena differences between the approximants. It is important to note that neither of these waveforms have termination conditions that are determined by the physical behavior of the inspiralling binary. The termination frequency only indicates the point at which the approximant is certainly no longer valid. The increased match for aligned spin waveforms is due to the higher frequency cutoff, which pushes the termination frequency out of the most sensitive part of the zero-detuned, high-power aLIGO sensitivity curve.

Figure 19 shows a comparison between the TaylorR2F4 and TaylorT4 models. In that figure, the second expansion associated with the TaylorR2F4 model is extended to order v^9 (left) and v^{12} (right), and the ending frequency of both is that corresponding to the MECO. We show that the TaylorR2F4 model is adequate for a large range of parameters as a computationally inexpensive substitute for TaylorT4.

Since the mismatch between the TaylorF2 and TaylorT4 models is not due to differences between the time domain and frequency domain approximants, this indicates that the effective higher order PN terms used in the construction of TaylorR2F4,

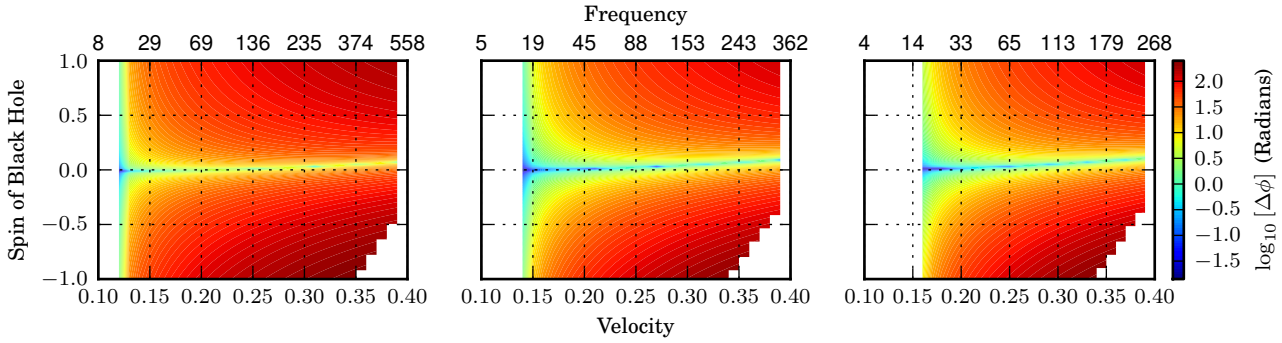

Figure 20: The accumulation of phase differences between TaylorT2 and TaylorT4, for systems with component masses (m_1, m_2) of $(1.4M_\odot, 6M_\odot)$ (left), $(1.4M_\odot, 10M_\odot)$ (center), and $(1.4M_\odot, 14M_\odot)$ (right). The approximants include spin terms up to 2.5PN. The calculation starts from the velocity corresponding to a gravitational-wave frequency of 15Hz, continues to the velocity on the horizontal axis, and reports the difference in accumulated gravitational-wave phase between the waveforms. The feature in the bottom right corner of each plot arises because the TaylorT2 approximant is no longer monotonic. Note that large phase differences accumulate at very low velocities $v \sim 0.2$ for even small black hole spins.

which are also intrinsically present in TaylorT4, are still significant. To obtain better agreement between the different PN approximants we consider, it is necessary to extend the PN expansions of the energy and flux equations to include unknown higher order terms, particularly ones that involve the spin of the objects.

3.8 Accumulation of Phase Discrepancy

In the previous sections, we demonstrated that the two PN approximants, TaylorF2 and TaylorT4, and the SEOBNRv1 model are not faithful to each other. We also showed that this is not due to the differences between frequency and time domain waveforms. From the construction of the TaylorR2F4 approximant, we also demonstrated that the two PN families can be written in a way that is consistent up to the chosen PN order, but where TaylorR2F4 contains higher order in v corrections that account for the differences between the models. Since these are higher order corrections, they should start to become important to the orbital phasing only at high velocities, and thus high gravitational-wave frequencies. In this section we investigate where, for systems with parameters corresponding to NSBH binaries, the approximants diverge.

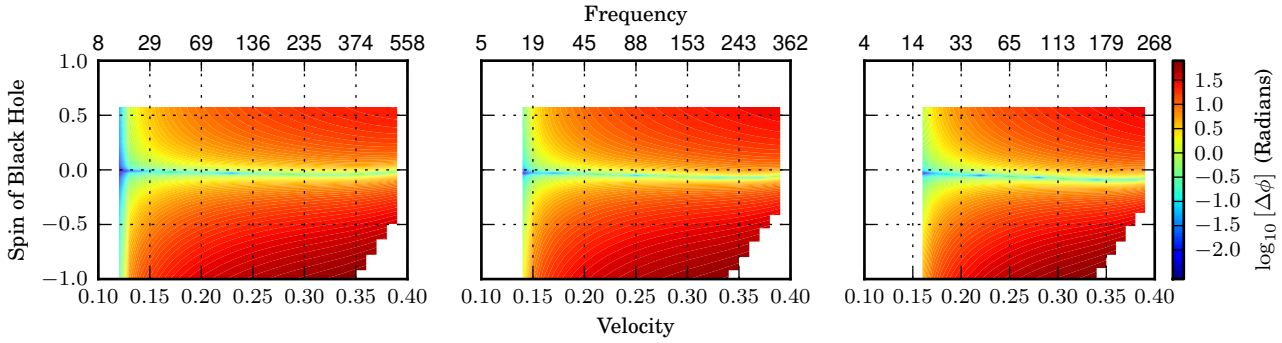

Figure 21: The accumulation of phase difference between TaylorT2 and SEOBNRv1, for systems with component masses (m_1, m_2) of $(6M_\odot, 1.4M_\odot)$ (left), $(10M_\odot, 1.4M_\odot)$ (center), and $(14M_\odot, 1.4M_\odot)$ (right). TaylorT2 includes spin terms up to 2.5PN. The calculation starts from the velocity corresponding to a gravitational-wave frequency of 15Hz, continues to the velocity on the horizontal axis, and reports the difference in accumulated gravitational-wave phase between the waveforms. The feature in the bottom right corner of each plot arises because the TaylorT2 approximant is no longer monotonic. As in Fig. 20, a large phase difference is accumulated at low velocities and small black hole spins.

We do this by examining the accumulation of phase as a function of orbital velocity and reporting the difference in the number of gravitational-wave cycles between different approximants.

In Fig. 20, we examine the difference in the accumulated phase between TaylorT2 and TaylorT4 for three example systems with component masses (m_1, m_2) of $(6M_\odot, 1.4M_\odot)$, $(10M_\odot, 1.4M_\odot)$, and $(14M_\odot, 1.4M_\odot)$. We see that the phase difference between the two models quickly grows to tens of radians, even when the black hole spin magnitude is small. This is also true when comparing TaylorT2 and SEOBNRv1, as can be seen in Fig. 21. In the latter case, there is also a noticeable deviation away from the line of zero spin where for unknown reasons the two models diverge and subsequently converge.

3.9 Accumulation of mismatch

As gravitational-wave detectors are not directly sensitive to phase differences alone, it is useful to compute how the match, which incorporates the sensitivity of a gravitational-wave detector, changes as a function of the upper frequency cutoff used for the calculation. In this section we demonstrate at which frequencies and corresponding

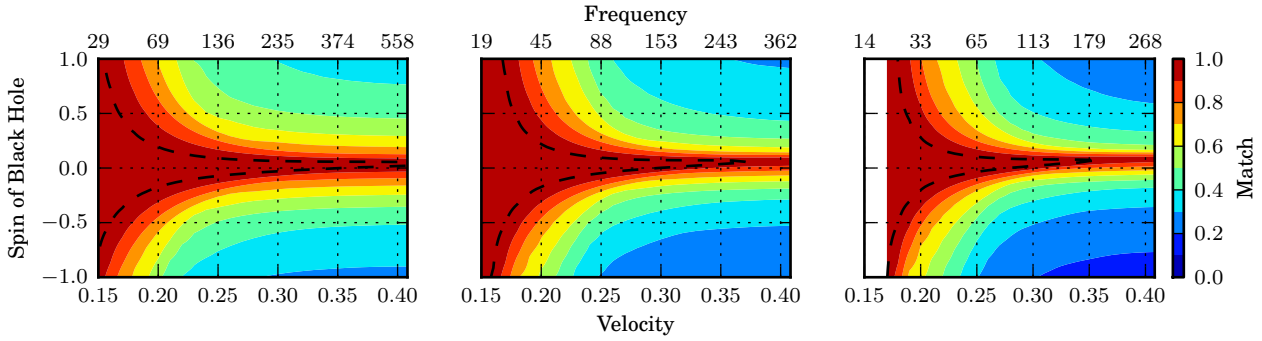

Figure 22: The match between TaylorF2 and TaylorT4 integrated from 15 Hz up to the designated frequency for systems with component masses (m_1, m_2) of $(1.4M_\odot, 6M_\odot)$ (left), $(1.4M_\odot, 10M_\odot)$ (center), and $(1.4M_\odot, 14M_\odot)$ (right). Both approximants include spin corrections up to 2.5PN. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve. A contour at a match of 0.97 is indicated by the dotted line. The match follows the general features seen in the phase difference comparison of Fig. 20 and drops significantly, even at relatively low velocities. For the $(1.4M_\odot, 6M_\odot)$ system with a black hole spin $\chi = 0.5$, the match has already dropped to ~ 0.5 at a velocity of only ~ 0.25.

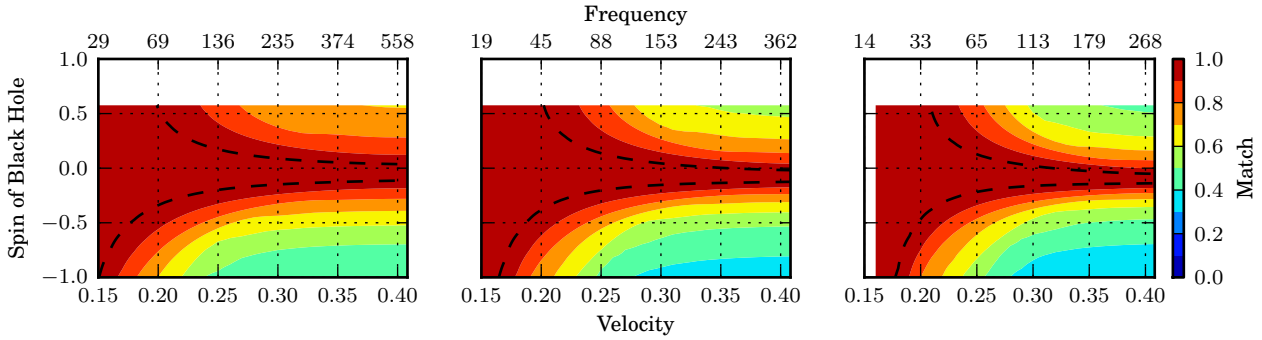

Figure 23: The match between the TaylorF2 and SEOBNRv1 models integrated from 15 Hz up to the designated frequency for systems with component masses (m_1, m_2) of $(6M_\odot, 1.4M_\odot)$ (left), $(10M_\odot, 1.4M_\odot)$ (center), and $(14M_\odot, 1.4M_\odot)$ (right). TaylorF2 includes spin corrections up to 2.5PN. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve. A contour at a match of 0.97 is indicated by the dotted line. We note that, although the match is marginally improved compared to Fig. 22, there are still large disagreements at velocities as low as 0.25.

velocities the match between waveform families drops. To do so, we define an inner product between waveforms that is a function of the upper frequency cutoff. This inner product is then used in the match calculation of Eq. (3.19).

In Fig. 22, we examine the match between TaylorF2 and TaylorT4, integrated from a lower frequency cutoff of 15 Hz up to the upper frequency cutoff indicated on the horizontal axis. This is compared for the same three example systems as in Sec. 3.8. The match is shown across the range of allowable values of the black hole spin and the neutron star spin is set to zero. We see that the match drops precipitously even at low velocities and relatively modest spin magnitudes. For example, for a system with $m_1 = 6M_\odot$, $m_2 = 1.4M_\odot$, and a dimensionless spin of 0.5 for the black hole, the match drops below 0.7 at a velocity of only 0.23. The loss in match is more pronounced with increasing mass ratio.

In Fig. 23, we examine the match between TaylorF2 and SEOBNRv1, integrated from a lower frequency cutoff of 15Hz up to the upper frequency cutoff indicated on the horizontal axis. Again, the match drops for large spin magnitudes at relatively low velocities, although, just as the TaylorF2 approximant has shown better matches with the SEOBNRv1 approximant than with the TaylorT4 approximant, this occurs at somewhat higher velocities. This shows clearly that significant portions of the loss in match seen in Sec. 3.5 occurs at unexpectedly low velocities.

3.10 Detection searches and Early aLIGO

In the previous sections, we have demonstrated a substantial loss in match between different PN and EOB models of NSBH binaries. These discrepancies will cause substantial biases in attempts to measure the parameters of detected systems with aLIGO. However, when detecting systems the *fitting factor*, rather than the match, is the quantity that is used to assess the effectualness of a search [**?**]. The fitting factor maximizes the match between a signal and a bank of templates designed to capture e.g. 97% of the optimal signal-to-noise ratio. The template bank is constructed to be valid for the same range of masses and spins used throughout this paper and detailed in Sec 6.1. Discrepancies in match due to differing approximants may be compensated for by allowing a waveform to match to a template with shifted parameters. Figs. 24 and 25 show the fitting factor of a TaylorF2 aligned spin template bank when used

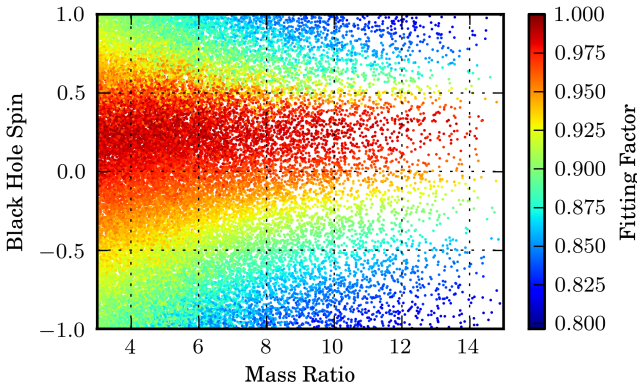

Figure 24: The fitting factor between the TaylorF2 and TaylorT4 approximants as a function of the spin of the black hole and the mass ratio of the system, when maximizing the match over a bank of TaylorF2 waveforms. All approximants include spin corrections up to 2.5PN. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve and a 15Hz lower frequency cutoff. In comparison to the match of these approximants shown in Fig. 13, we see that while allowing for the maximization over a bank of templates has improved the overall agreement, it is unable to entirely make up for the poor match.

to detect TaylorT4 waveforms. Fig. 24 shows the distribution of fittings factors for approximants that include up to the 2.5PN spin corrections. Fig. 25 demonstrates the effect of adding the higher order 3.0PN spin-orbit tail and 3.5PN spin-orbit corrections. Construction of these aligned spin banks use the method introduced in Ref. [**?**] and is described in more detail in Ref. [**?**]. There is substantial improvement in the fitting factors of aligned spin systems when adding the higher order spin corrections, but no improvement for anti-aligned spin systems. Although the loss in fitting factor is not as significant as the loss in match shown in Figs. 13 and 14, aLIGO NSBH searches will isncur a substantial loss in signal-to-noise ratio for anti-aligned spins, if the accuracy of NSBH waveforms is not improved.

In the previous sections we have modeled the sensitivity of aLIGO with the zero-detuned, high-power sensitivity curve [**?**]. Early commissioning scenarios for aLIGO indicate that observations will begin with less sensitivity in the 10–40 Hz region [**?**]. We investigate if the substantial disagreement found between TaylorF2 and TaylorT4 is still present for early detector sensitives by a instead using a lower frequency cutoff of 30 Hz.

In Fig. 26 and 27, we show the faithfulness between the TaylorF2 and TaylorT4

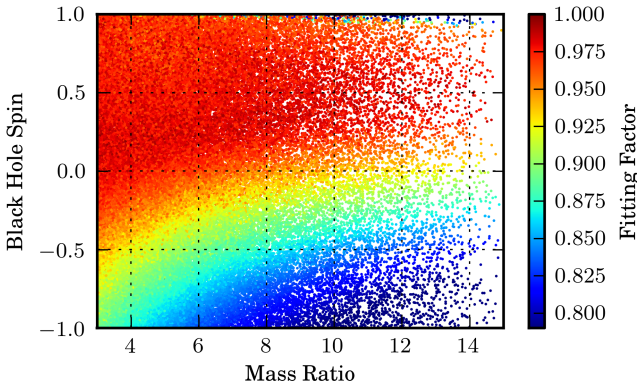

Figure 25: The fitting factor between the TaylorF2 and TaylorT4 approximants as a function of the spin of the black hole and the mass ratio of the system, when maximizing the match over a bank of TaylorF2 waveforms. All approximants include the 3.5PN spin-orbit and 3.0PN spin-orbit tail corrections. Matches are calculated using the the aLIGO zero-detuned, high-power sensitivity curve and a 15Hz lower frequency cutoff. In comparison to the fitting factors shown in Fig. 24, we see that adding the higher order spin corrections has resulted in substantially improved fitting factors for systems where the spin is aligned with the orbital angular momentum. There is no improvement for anti-aligned systems.

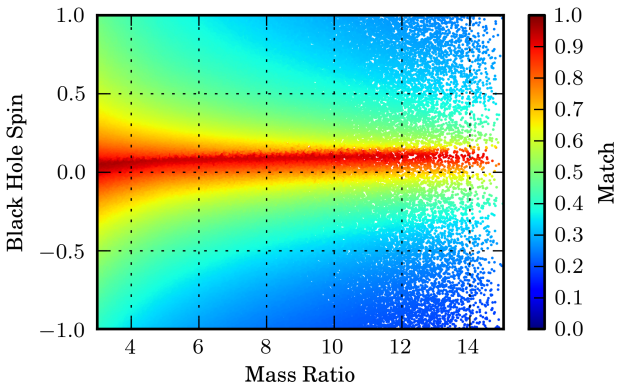

Figure 26: The match between TaylorF2 and TaylorT4 as a function of the spin of the black hole and the mass ratio of the system. The approximants include spin corrections up to 2.5PN. Matches are calculated using a 30Hz lower frequency cutoff to approximate the sensitivity of an early aLIGO detector. In comparison to Fig. 13, which uses a 15Hz lower frequency cutoff, there is only a negligible improvement in match. Matches remain low at moderate black hole spins $\chi \sim 0.3$.

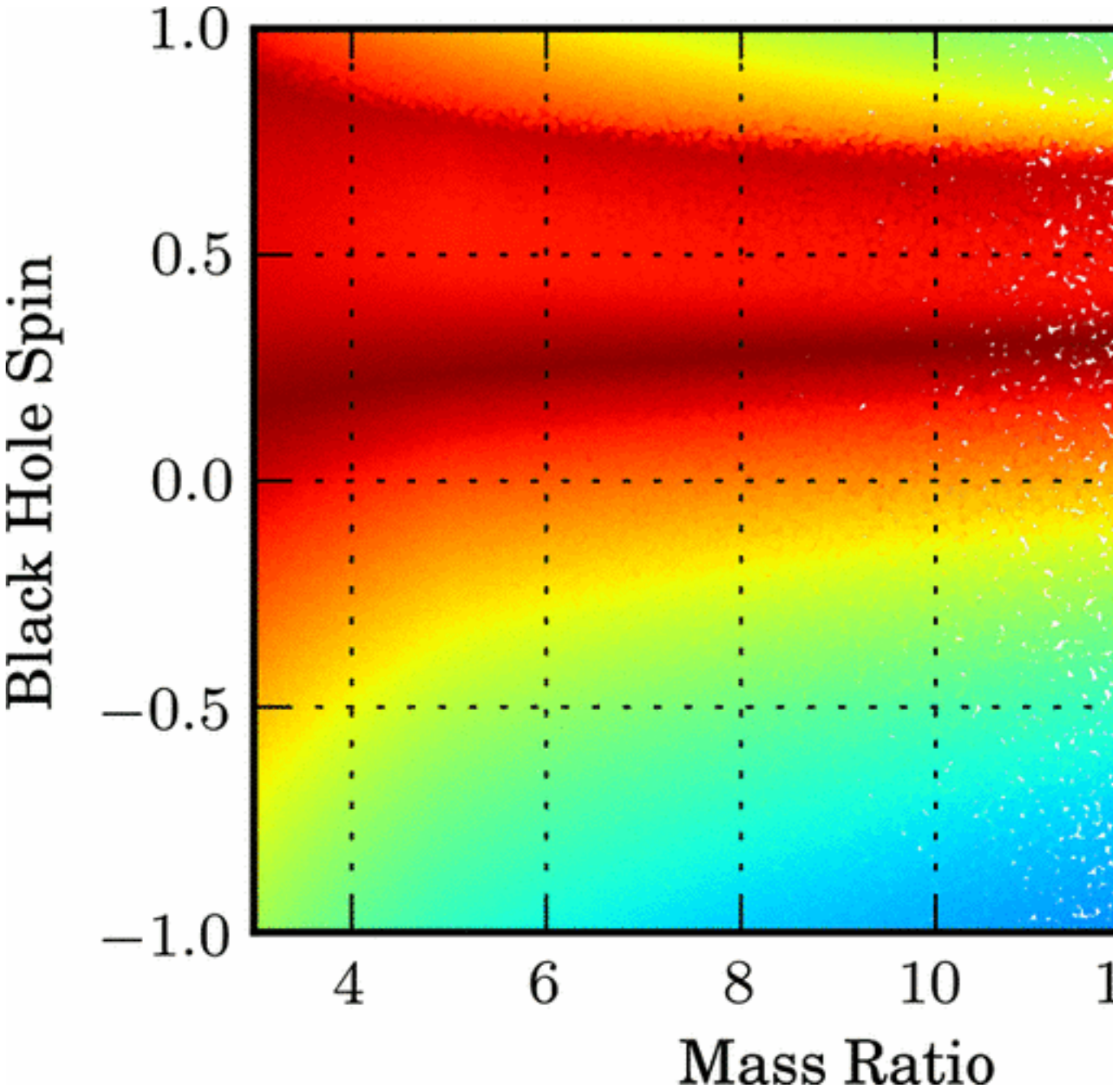

Figure 27: The match TaylorF2 and TaylorT4 approximants, with the 3.5PN spin-orbit and 3.0PN spin-orbit tail corrections included, as a function of the spin of the black hole and the mass ratio of the system. The approximants include only the nown spin terms up to 2.5PN. Matches are calculated using a 30Hz lower frequency cutoff to approximate the sensitivity of the early aLIGO detector. In comparison to Fig. 14, which uses a 15Hz lower frequency cutoff, there is only a negligible improvement in match.

approximants that include only the complete 2.5 PN and partial 3.5PN spin-related corrections, respectively. We see that there is no significant improvement in the faithfulness of the approximants, and so additional spin corrections are desirable even for early detector scenarios.

3.11 Conclusions

We have found that there is significant disagreement between NSBH waveforms modelled with the TaylorT2, TaylorT4, and SEOBNRv1 approximants. This will pose problems for the construction of optimal NSBH detection searches, potentially reducing the event rate, and may cause significant biases in the parameter measurement of detected signals.

The discrepancies are not accounted for by the differences between frequency and time domain waveforms and start at fairly low ($v \sim 0.2$) orbital velocities. Since the discrepancies in the approximants result from how the PN expansions of the energy and flux are combined and truncated, we conclude that the calculation of higher order PN terms is required to increase the faithfulness of these approximants, and more importantly, to improve the ability to detect NSBH coalescences. The discrepancies between approximants are significantly smaller when the spin of the black hole is close to zero, which further motivates the calculation of the PN terms associated with the spin of the objects beyond those known completely up to 2.5PN order and partially up to 3.5PN. Therefore, additional work is needed to verify the validity of waveform models used for NSBH searches. We also note that we have only compared different waveform families under the assumption that the spins of the component objects are (anti-)aligned with the orbital angular momentum of the system. It is expected that generic NSBH systems will not be limited to aligned spins, but may instead be more isotropically oriented. This could lead to an additional source of discrepancy between our models and the true signal, which would result in an additional loss in the detection rate of sources.

Acknowledgements

We thank Stefan Ballmer, Alessandra Buonanno, Eliu Huerta, Prayush Kumar, Richard O'Shaughnessy, B. S. Sathyaprakash, Peter Saulson, and Matt West for useful discussions. This work is supported by National Science Foundation awards PHY-0847611 (DAB, AHN), PHY-1205835 (AHN, IWH), PHY-0970074 (EO), and PHY-0855589 (AL). DAB, IWH, AL, and EO thank the Kavli Institute for Theoretical Physics at Santa Barbara University, supported in part by NSF grant PHY-0551164, for hospitality during this work. DAB thanks the LIGO Laboratory Visitors Program, supported by NSF cooperative agreement PHY-0757058, for hospitality. DK and AL thank the Max Planck Gesellschaft for support. DAB is supported by a Cottrell Scholar award from the Research Corporation for Science Advancement. Computations used in this work were performed on the Syracuse University Gravitation and Relativity cluster, which is supported by NSF awards PHY-1040231 and PHY-1104371.

3.12 Post-Newtonian Energy and Gravitational-wave Flux

In this appendix, we give the PN coefficients for the center of mass energy E_i and the gravitational-wave flux F_i, whose contributions were derived and presented in [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**]. We include corrections that involve the component objects' spins up to 3.5PN. These coefficients depend on the dimensionless spins of the component objects $\chi_i = \mathbf{S}_i/m_i^2$, their projections onto the direction of so-called Newtonian orbital angular momentum $\mathbf{L}_N = M\eta\mathbf{r}\times\dot{\mathbf{r}}$, and the symmetric mass ratio η. Additionally, quadrupole-monopole contributions depend on a parameter q_i, which quantifies the strength of the quadrupole moment induced by the oblateness of each spinning compact object. For BHs, $q_i = 1$, while for NSs q_i will depend on the equation of state, with [**?**] finding $q_i \sim 2 - 12$.

The coefficients associated with the energy are given as

$$E_{\rm Newt} = -\frac{M}{2}\eta, \tag{3.24}$$

$$E_2 = -\frac{3}{4} - \frac{1}{12}\eta, \tag{3.25}$$

$$E_3 = \sum_{i=1}^{2} \left[\frac{8}{3} \left(\frac{m_i}{M} \right)^2 + 2\eta \right] \left(\chi_i \cdot \hat{L}_N \right), \tag{3.26}$$

$$E_4 = -\frac{27}{8} + \frac{19}{8}\eta - \frac{1}{24}\eta^2 + \eta \left[\chi_1 \cdot \chi_2 - 3 \left(\chi_1 \cdot \hat{L}_N \right) \left(\chi_2 \cdot \hat{L}_N \right) \right] + \frac{1}{2} \sum_{i=1}^{2} q_i \left(\frac{m_i}{M} \right)^2 \left[\chi_i^2 - 3 \left(\chi_i \cdot \hat{L}_N \right)^2 \right], \tag{3.27}$$

$$E_5 = \sum_{i=1}^{2} \left[\left(8 - \frac{31}{9}\eta \right) \left(\frac{m_i}{M} \right)^2 + \eta \left(3 - \frac{10}{3}\eta \right) \right] \left(\chi_i \cdot \hat{L}_N \right), \tag{3.28}$$

$$E_6 = -\frac{675}{64} + \left(\frac{34\,445}{576} - \frac{205}{96}\pi^2 \right) \eta - \frac{155}{96}\eta^2 - \frac{35}{5184}\eta^3, \tag{3.29}$$

$$E_7 = \sum_{i=1}^{2} \left[\left(27 - \frac{211}{4}\eta + \frac{7}{6}\eta^2 \right) \left(\frac{m_i}{M} \right)^2 + \eta \left(\frac{27}{4} - 39\eta + \frac{5}{4}\eta^2 \right) \right] \left(\chi_i \cdot \hat{L}_N \right). \tag{3.30}$$

The coefficients associated with the flux are given as

$$F_{\rm Newt} = \frac{32}{5}\eta^2, \tag{3.31}$$

$$F_2 = -\frac{1247}{336} - \frac{35}{12}\eta, \tag{3.32}$$

$$F_3 = 4\pi - \sum_{i=1}^{2} \left[\frac{11}{4} \left(\frac{m_i}{M} \right)^2 + \frac{5}{4}\eta \right] \left(\chi_i \cdot \hat{L}_N \right), \tag{3.33}$$

$$\begin{aligned} F_4 &= -\frac{44\,711}{9072} + \frac{9271}{504}\eta + \frac{65}{18}\eta^2 + \eta \left[\frac{289}{48} \left(\chi_1 \cdot \hat{L}_N \right) \left(\chi_2 \cdot \hat{L}_N \right) - \frac{103}{48} \chi_1 \cdot \chi_2 \right] \\ &+ \sum_{i=1}^{2} q_i \left(\frac{m_i}{M} \right)^2 \left[3 \left(\chi_i \cdot \hat{L}_N \right)^2 - \chi_i^2 \right] + \frac{1}{96} \left(\frac{m_i}{M} \right)^2 \left[7\chi_i^2 - \left(\chi_i \cdot \hat{L}_N \right)^2 \right], \end{aligned} \tag{3.34}$$

$$F_5 = \left(-\frac{8191}{672} - \frac{583}{24}\eta \right) \pi + \sum_{i=1}^{2} \left[\left(-\frac{59}{16} + \frac{701}{36}\eta \right) \left(\frac{m_i}{M} \right)^2 + \eta \left(-\frac{13}{16} + \frac{43}{4}\eta \right) \right] \left(\chi_i \cdot \hat{L}_N \right), \tag{3.35}$$

$$\begin{aligned} F_6 &= \frac{6\,653\,739\,519}{69\,854\,400} + \frac{16}{3}\pi^2 - \frac{1712}{105}\gamma_E - \frac{856}{105}\log(16v^2) + \left(-\frac{134\,543}{7776} + \frac{41}{48}\pi^2 \right) \eta \\ &- \frac{94\,403}{3024}\eta^2 - \frac{775}{324}\eta^3 - \pi \sum_{i=1}^{2} \left[\frac{65}{6} \left(\frac{m_i}{M} \right)^2 - \frac{31}{6}\eta \right] \left(\chi_i \cdot \hat{L}_N \right), \end{aligned} \tag{3.36}$$

$$
\begin{aligned}
F_7 &= \left(-\frac{16\,285}{504}+\frac{214\,745}{1728}\eta+\frac{193\,385}{3024}\eta^2\right)\pi \\
&+ \sum_{i=1}^{2}\left[\left(\frac{162\,035}{3888}+\frac{971}{54}\eta-\frac{6737}{108}\eta^2\right)\left(\frac{m_i}{M}\right)^2+\eta\left(\frac{9535}{336}+\frac{1849}{126}\eta-\frac{1501}{36}\eta^2\right)\right]\left(\chi_i\cdot\hat{L}_N\right)
\end{aligned}
\tag{3.37}
$$

3.13 Post-Newtonian Approximants

The PN approximants TaylorT4, TaylorT2, TaylorF2, and TaylorR2F4 are given using the flux up to 3.5 PN and the center-of-mass energy up to 3.0 PN. Corrections due to spin are included up to 3.5 PN order. This includes the leading order spin orbit correction β at 1.5PN, leading order spin-spin correction σ at 2PN (which includes quadrupole-monopole and so-called self-spin effects proportional to s_i^2), next-to-leading order spin-orbit corrections γ at 2.5PN, tail-induced spin orbit correction ξ at 3PN, and third order spin-orbit correction ζ appearing at 3.5PN. These corrections can be expressed as,

$$
\beta=\sum_{i=1}^{2}\left[\frac{113}{12}\left(\frac{m_i}{M}\right)^2+\frac{25}{4}\eta\right]\left(\chi_i\cdot\hat{L}_N\right), \tag{3.38}
$$

$$
\begin{aligned}
\sigma=\eta&\left[\frac{721}{48}\left(\chi_1\cdot\hat{L}_N\right)\left(\chi_2\cdot\hat{L}_N\right)-\frac{247}{48}\chi_1\cdot\chi_2\right] \\
&+\frac{5}{2}\sum_{i=1}^{2}q_i\left(\frac{m_i}{M}\right)^2\left[3\left(\chi_i\cdot\hat{L}_N\right)^2-\chi_i^2\right] \\
&+\frac{1}{96}\sum_{i=1}^{2}\left(\frac{m_i}{M}\right)^2\left[7\chi_i^2-\left(\chi_i\cdot\hat{L}_N\right)^2\right],
\end{aligned}
\tag{3.39}
$$

$$
\gamma=\sum_{i=1}^{2}\left[\left(\frac{732\,985}{2268}+\frac{140}{9}\eta\right)\left(\frac{m_i}{M}\right)^2+\eta\left(\frac{13\,915}{84}-\frac{10}{3}\eta\right)\right]\left(\chi_i\cdot\hat{L}_N\right), \tag{3.40}
$$

$$
\xi=\pi\sum_{i=1}^{2}\left[\frac{75}{2}\left(\frac{m_i}{M}\right)^2+\frac{151}{6}\eta\right]\left(\chi_i\cdot\hat{L}_N\right), \tag{3.41}
$$

$$\zeta = \sum_{i=1}^{2} \left[\left(\frac{130\,325}{756} - \frac{796\,069}{2016}\eta + \frac{100\,019}{864}\eta^2 \right) \left(\frac{m_i}{M} \right)^2 + \eta \left(\frac{1\,195\,759}{18\,144} - \frac{257\,023}{1008}\eta + \frac{2903}{32}\eta^2 \right) \right] \left(\chi_i \cdot \hat{L}_{\text{I}} \right.$$

3.13.1 TaylorT4

$$\begin{aligned} \frac{dv}{dt} = \frac{32\eta}{5M} v^9 \Bigg\{ 1 \;&+\; \left(-\frac{743}{336} - \frac{11}{4}\eta \right) v^2 + (4\pi - \beta) v^3 + \left(\frac{34\,103}{18\,144} + \frac{13\,661}{2016}\eta + \frac{59}{18}\eta^2 + \sigma \right) v^4 \\ &+\; \left(-\frac{4159\pi}{672} - \frac{189\pi}{8}\eta - \frac{9}{40}\gamma + \left(\frac{743}{168} + \frac{11}{2}\eta \right) \beta \right) v^5 \\ &+\; \Bigg[\frac{16\,447\,322\,263}{139\,708\,800} - \frac{1712\gamma_E}{105} + \frac{16\pi^2}{3} - \frac{1712}{105} \log(4v) \\ &\qquad + \left(-\frac{56\,198\,689}{217\,728} + \frac{451\pi^2}{48} \right) \eta + \frac{541}{896}\eta^2 - \frac{5605}{2592}\eta^3 - \xi \Bigg] v^6 \\ &+\; \pi \left(-\frac{4415}{4032} + \frac{358\,675}{6048}\eta + \frac{91\,495}{1512}\eta^2 - \zeta \right) v^7 \Bigg\} \end{aligned} \tag{3.43}$$

3.13.2 TaylorT2

$$\begin{aligned} \frac{dt}{dv} = \frac{5M}{32\eta} v^{-9} \Bigg\{ 1 \;&+\; \left(\frac{743}{336} + \frac{11}{4}\eta \right) v^2 + (-4\pi + \beta) v^3 + \left(\frac{3\,058\,673}{1\,016\,064} + \frac{5429}{1008}\eta + \frac{617}{144}\eta^2 - \right. \\ &+\; \left(-\frac{7729\pi}{672} + \frac{13\pi}{8}\eta + \frac{9}{40}\gamma \right) v^5 \\ &+\; \Bigg[-\frac{10\,817\,850\,546\,611}{93\,884\,313\,600} + \frac{32\pi^2}{3} + \frac{1712\gamma_E}{105} + \frac{1712}{105} \log(4v) \\ &\qquad + \left(\frac{3\,147\,553\,127}{12\,192\,768} - \frac{451\pi^2}{48} \right) \eta - \frac{15\,211}{6912}\eta^2 + \frac{25\,565}{5184}\eta^3 - 8\pi\beta + \delta \\ &+\; \pi \Bigg(-\frac{15\,419\,335}{1\,016\,064} - \frac{75\,703}{6048}\eta + \frac{14\,809}{3024}\eta^2 - \beta \left(\frac{8\,787\,977}{1\,016\,064} + \frac{51\,841}{2016}\eta \right. \right. \end{aligned}$$

$$\left. \left. + \gamma \left(\frac{2229}{2240} + \frac{99}{80}\eta \right) + \zeta \right) v^7 \Bigg\}$$

3.13.3 TaylorF2

$$A_{(F2)}(f) \propto \frac{(\pi M f)^{2/3}}{\sqrt{\dot{F}(f)}} \tag{3.45}$$

$$\begin{aligned}
\psi_{(F2)}(f) &= 2\pi f t_c - \phi_c + \frac{3}{128\eta} v^{-5} \Bigg\{ 1 + \left(\frac{3715}{756} + \frac{55}{9}\eta \right) v^2 + (4\beta - 16\pi) v^3 \\
&+ \left(\frac{15\,293\,365}{508\,032} + \frac{27\,145}{504}\eta + \frac{3085}{72}\eta^2 - 10\sigma \right) v^4 + \left(\frac{38\,645}{756}\pi - \frac{65}{9}\pi\eta - \gamma \right) (1 + 3\log(v))\, v^5 \\
&+ \Bigg[\frac{11\,583\,231\,236\,531}{4\,694\,215\,680} - \frac{640}{3}\pi^2 - \frac{6848\gamma_E}{21} - \frac{6848}{21}\log(4v) + \left(-\frac{15\,737\,765\,635}{3\,048\,192} + \frac{2255\pi^2}{12} \right) \\
&+ \frac{76\,055}{1728}\eta^2 - \frac{127\,825}{1296}\eta^3 + 160\pi\beta - 20\xi \Bigg] v^6 + \pi \left(\frac{77\,096\,675}{254\,016} + \frac{378\,515}{1512}\eta - \frac{74\,045}{756}\eta^2 \right. \\
&+ \left. \beta \left(\frac{43\,939\,885}{254\,016} + \frac{259\,205}{504}\eta + \frac{10\,165}{36}\eta^2 \right) - \gamma \left(\frac{2229}{112} - \frac{99}{4}\eta \right) - 20\zeta \right) v^7 \Bigg\}
\end{aligned} \tag{3.4}$$

3.13.4 TaylorR2F4

In the equation below, the a_i are the PN coefficients of the TaylorT4 expansion

$$\left(\frac{dv}{dt} \right)_{T4} = A_7(v) = a_0 \left(1 + \sum_{i=2}^{7} a_i v^i + a_{6\log} v^6 \log(4v) \right).$$

which can be read off of Eq. (3.43).

$$
\begin{aligned}
\psi_{(R2F4)}(f) &= \psi_{(F2)}(f) + \frac{3}{128\eta}v^{-5}\Bigg\{\Bigg[-20\beta^2+\sigma\left(\frac{3715}{42}+100\eta\right)\Bigg]v^6+\Bigg[(40\beta-160\pi)\,\sigma\Bigg]v^7 \\
&+ \frac{40}{9}\Bigg[\left(3a_2^2a_4-a_2^4-a_4^2-2a_3a_5+3a_2a_3^2-2a_2a_6+\frac{2}{3}a_2a_{6\log}\right)(1-3\log(v)) \\
&+ 3a_2a_{6\log}\log(4v)^2\Bigg]v^8+5\Bigg[8a_2^2a_3-2a_3^3-12a_2a_3a_4-6a_2^2a_5+4a_4a_5+4a_3a_6-5a_3a_{6\log} \\
&+ 4a_2a_7+4a_3a_{6\log}\log(4v)\Bigg]v^9+4\Bigg[-a_2^5+6a_2^2a_3^2+4a_2^3a_4-3a_3^2a_4-3a_2a_4^2-6a_2a_3a_5 \\
&+ a_5^2-3a_2^2a_6+2a_4a_6+2a_3a_7+\left(\frac{21}{10}a_2^2-\frac{7}{5}a_4\right)a_{6\log}+\left(2a_4a_{6\log}-3a_2^2a_{6\log}\right)\log(4v)\Bigg]v^{10} \\
&+ \frac{20}{9}\Bigg[-5a_2^4a_3+4a_2a_3^3+4a_2^3a_5+12a_2^2a_3a_4-3a_3a_4^2-3a_3^2a_5-6a_2a_4a_5-6a_2a_3a_6 \\
&+ 2a_5a_6+3a_2a_3a_{6\log}-a_5a_{6\log}+2a_4a_7-3a_2^2a_7+(2a_5-6a_2a_3)\,a_{6\log}\log(4v)\Bigg]v^{11} \\
&+ \frac{10}{7}\Bigg[a_2^6-10a_2^3a_3^2+a_3^4-5a_2^4a_4+12a_2a_3^2a_4+6a_2^2a_4^2-a_4^3+12a_2^2a_3a_5-6a_3a_4a_5-3a_2a_5^2 \\
&+ 4a_2^3a_6-3a_3^2a_6-6a_2a_4a_6+a_6^2+\left(-\frac{11}{7}a_2^3+\frac{33}{28}a_3^2+\frac{33}{14}a_2a_4-\frac{11}{14}a_6+\frac{93}{392}a_{6\log}\right)a_{6\log} \\
&+ 2a_5a_7+\left(4a_2^3-3a_3^2-6a_2a_4+2a_6-\frac{11}{14}a_{6\log}\right)a_{6\log}\log(4v)+a_{6\log}^2\log(4v)^2\Bigg]v^{12}\Bigg\}
\end{aligned}
$$

Chapter 4

NSBH Precession

4.1 abstract

The first direct detection of neutron-star–black-hole binaries will likely be made with gravitational-wave observatories. Advanced LIGO and Advanced Virgo will be able to observe neutron-star–black-hole mergers at a maximum distance of 900Mpc. To acheive this sensitivity, gravitational-wave searches will rely on using a bank of filter waveforms that accurately model the expected gravitational-wave signal. The emitted signal will depend on the masses of the black hole and the neutron star and also the angular momentum of both components. The angular momentum of the black hole is expected to be comparable to the orbital angular momentum when the system is emitting gravitational waves in Advanced LIGO's and Advanced Virgo's sensitive band. This angular momentum will affect the dynamics of the inspiralling system and alter the phase evolution of the emitted gravitational-wave signal. In addition, if the black hole's angular momentum is not aligned with the orbital angular momentum it will cause the orbital plane of the system to precess. In this work we demonstrate that if the effect of the black hole's angular momentum is neglected in the waveform models used in gravitational-wave searches, the detection rate of $(10 + 1.4)M_{\odot}$ neutron-star–black-hole systems would be reduced by $33 - 37\%$. The error in this measurement is due to uncertainty in the Post-Newtonian approximations that are used to model the gravitational-wave signal of neutron-star–black-hole inspiralling binaries. We describe a new method for creating a bank of filter waveforms where the black hole has non-zero angular momentum that is aligned with the orbital angular momentum. With this

bank we find that the detection rate of $(10+1.4)M_{\odot}$ neutron-star–black-hole systems would be reduced by $26-33\%$. Systems that will not be detected are ones where the precession of the orbital plane causes the gravitational-wave signal to match poorly with non-precessing filter waveforms. We identify the regions of parameter space where such systems occur and suggest methods for searching for highly precessing neutron-star–black-hole binaries.

4.2 Introduction

aLIGO will begin observing the gravitational-wave sky in 2015 [**?**]. When aLIGO reaches design sensitivity, it will be sensitive to a volume of the universe a thousand times greater than the first-generation LIGO detectors [**?**]. The French-Italian Advanced Virgo (AdV) detector will begin observations shortly after aLIGO, forming a world-wide network of gravitational-wave observatories [**?**, **?**, **?**]. One of the most interesting sources for aLIGO and AdV is the inspiral and merger of NSBH binaries. It has been argued that Cyg X-3 is a possible NSBH *progenitor* [**?**], however NSBH binaries have not been observed by radio or other electromagnetic observations. The first direct detection of a NSBH binary will likely be made with aLIGO and AdV. Population-synthesis models of binary evolution predict that aLIGO should see 0.2–300 NSBH binaries per year [**?**]. Direct detection of the gravitational waves from NSBH binaries would confirm their existence and allow us to explore the astrophysics behind the formation and evolution of these systems.

The gravitational waves radiated by NSBH binaries are expected to be significantly affected by the black hole's angular momentum (spin), which is expected to be comparable to the orbital angular momentum of the binary [**?**, **?**, **?**, **?**]. Spin-orbit coupling changes the gravitational waveform of the binary's inspiral and merger and can cause the orbital plane of the binary to precess [**?**]. Coupling between the black hole spin and the neutron star spin [**?**], the quadrupole-monopole interaction due to the spheroidal deformation of spinning black holes and neutron stars [**?**] and the "self-spin" interaction [**?**] will also effect the gravitational waveform emitted during a NSBH binary inspiral. The resulting changes in the waveform observed by aLIGO carry a great deal of information about the dynamics of the binary. However, optimal searches of aLIGO data must incorporate this dynamics into their waveform models

to avoid a reduction in sensitivity and hence the rate of detected events. Variation between the available waveform models, and with nature's waveforms, will also cause a reduction in sensitivity, we investigate this issue in a companion work [?].

Gravitational wave searches for the merger of two compact objects rely on matched-filtering against compact binary merger gravitational waveform models [?, ?, ?]. Compact binary mergers in quasi-circular orbit are described by 15 parameters; the masses, spin magnitude, spin orientations, source orientation, sky location, distance and time and phase of coalescence [?, ?]. Matched-filter searches must be capable of detecting binary mergers regardless of the parameters of the system. For non-precessing systems and restricting to the dominant gravitational wave mode, the extrinsic parameters - source orientation, sky location, distance and coalescence phase - only effect the overall phase and amplitude of the observed gravitational wave system. Therefore, it is possible to analytically maximize over these extrinsic parameters [?].

Changing the masses and spin magnitudes of a non-precessing system will change the intrinsic phase evolution of the system. To be able to detect NSBH systems within the desired parameter range a set of waveforms or "template bank" must be constructed [?, ?, ?, ?, ?, ?, ?]. These waveforms should span the desired range of mass and spins. The standard practice is to construct a bank of waveforms such that any waveform within the parameter space of interest would be recovered with at least 97% of the optimal SNR by at least one waveform in the template bank [?, ?] . However, the geometrical placement algorithms employed in the most recent searches for compact binary coalescences (CBCs) in LIGO and Virgo data are only appplicable for compact binary systems whose components have no angular momentum—non-spinning systems [?, ?, ?, ?]. Stochastic placement algorithms [?, ?, ?, ?] are capable of placing banks of waveforms where the spin of the black hole is aligned with the orbital angular momentum (aligned-spin NSBH) [?]. However, these algorithms are known to need more templates to cover a parameter space when compared to geometric algorithms [?]. In [?] we developed a new geometrical placement algorithm that could place template banks of aligned-spin BNS signals. In this work we expand that method to be able to place template banks of aligned-spin NSBH signals.

When precessing systems are considered as template waveforms, the matched-filter search becomes more complex. In this case the extrinsic parameters no longer enter as overall phase and amplitude shifts in the waveform [?]. Previous work has been

conducted to explore the affect of precession on gravitational-wave searches and to develop methods to detect precessing systems [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]. However, these searches, when applied to Initial LIGO and Virgo data, have not shown an increase in efficiency with respect to non-precessing searches [?]. This is because the filtering codes allow for increased, and unphysical, freedom when maximizing over extrinsic parameters and because no suitable method to distinguish gravitational wave signals from non-Gaussian instrumental noise has been developed for these searches. Therefore, searches for NSBH binaries in data from LIGO and Virgo's most recent science runs ignored spin affects and used quasi-circular templates to search for NSBH binaries [?, ?, ?, ?].

The majority of previous work considered the Initial LIGO detectors. aLIGO will have a substantially different noise curve than Initial LIGO [?]. Conclusions drawn using the Initial LIGO sensitivity curve may not hold when considering aLIGO. A previous study considering aLIGO sensitivity curves has suggested that it may be possible to detect generic, precessing NSBH binaries using aligned-spin waveforms [?]. However, other studies have suggested that precession may significantly change the gravitational waveform seen by aLIGO, requiring templates that explicitly capture this effect [?].

In this paper, we first investigate the effect of ignoring spin on optimal (matched-filter) searches for NSBH binaries with aLIGO. We demonstrate that the quasi-circular templates used in Initial LIGO will reduce the detection rate by $33-37\%$ for NSBH systems with masses uniformly distributed between $(10 \pm 0.5, 1.4 \pm 0.05)M_{\odot}$, an isotropic black hole spin distribution and spin magnitude uniformly distributed between 0 and 1. Over a wider range of uniformly distributed masses, $(3-15, 1-3)M_{\odot}$, we find that the detection rate would be reduced by $31-36\%$. In both cases this loss in detection rate is compared against a template bank where every signal is matched exactly by the bank of filters. The loss in event rate is greatest for NSBH binaries with large black-hole spins and large mass ratios. The range quoted in both measurements is due to uncertainty in the waveform models used to simulate NSBH gravitational-wave signals. These values also strongly depend on the signal distributions that we selected. If nature does not provide a uniform distribution of masses and an isotropic distribution of masses then these averaged values will change. To account for this, we explore the ability to recover NSBH signals as a function of their spins and masses in

section 4.9.

We expand upon the method we introduced in [?] and construct a bank of templates for aligned-spin NSBH binaries. We demonstrate that this template bank is effectual for recovering the population of aligned-spin NSBH systems that it is designed to detect. We assess the ability of an aligned-spin template bank to detect a population of generic NSBH binaries where the black hole spin is not constrained to be parallel to the orbital angular momentum. We find using the aligned-spin bank will reduce the detection rate by $17 - 23\%$ compared to using a bank where every signal matches exactly with one of the filter waveforms when searching for NSBH waveforms with masses $(3 - 15, 1 - 3)M_{\odot}$. When restricting the mass range to $(10 \pm 0.5, 1.4 \pm 0.05)M_{\odot}$ we find that the detection rate is reduced by $26 - 33\%$. We find that there are regions of the NSBH signal parameter space where precession effects cause a significant reduction in signal-to-noise ratio. These regions are those where the black hole's angular momentum is large in comparison to the orbital angular momentum. We suggest possible methods for constructing searches that recover these systems. By considering several NSBH waveform models, we demonstrate that our results are robust against possible errors in the post-Newtonian phasing for NSBH binaries.

There has been a great deal of recent work focused on numerically modelling the merger of a black hole and a neutron star [?, ?, ?, ?, ?]. However, there is not currently any widely available waveform model that includes both the full evolution of a NSBH coalescence *and* includes precessional effects over the full parameter space that we consider. Therefore, in this work we have restricted ourselves to considering post-Newtonian, inspiral-only signal waveforms and consider only the case of two point particles. If a full inspiral-merger-ringdown, precessing NSBH waveform model becomes available, it would be informative to compare results with that model against those presented here. However, in this work the black hole mass is restricted to be less than $15M_{\odot}$. It has been demonstrated that inspiral-only template banks recover $> 95\%$ of the signal power of numerically modelled $(3 + 15)M_{\odot}$ binary black hole waveforms [?, ?]. It has also been demonstrated that non-spinning NSBH mergers with total mass $\sim 10M_{\odot}$ are indistinguishable from binary black holes (BBH) mergers with the same masses [?]. With these observations we expect that our results are qualitatively valid in the parameter space we study.

The layout of this work is as follows. In section 4.3 we describe the set of NSBH systems that we use to assess the performance of our template banks. In section 4.4 we discuss the waveform models that we use in our simulations. In section 4.5 we discuss the methods we use to test the template banks. In section 4.6 we describe our new method to create banks of aligned-spin filter waveforms and use these methods in section 4.7 to create our template banks. In section 4.8 we validate our template banks against the aligned-spin signal models they are constructed to detect. In section 4.9 we assess the performance of non-spinning template banks to search for generic NSBH signals and assess the performance of aligned-spin template banks to detect the same signals. We conclude in section 4.10. Throughout this work we will use $G = c = 1$.

4.3 A population of NSBH binaries

In this section, we describe our large simulated set of NSBH binaries. This is used to assess the loss in detection rate when using non-spinning and aligned-spin template banks to search for generic NSBH binaries. To construct this set we incorporate current astrophysical knowledge to choose the distribution of masses and spins. However, this astrophysical knowledge is limited due to the fact that no NSBH binaries have been directly observed. Nevertheless, both NSs and BHs have been observed in other binary systems, and these observations can be used to make inferences about the mass and spin distributions that might be expected in NSBH binaries. We begin by giving the distributions that we use in this work, before describing the astrophysical knowledge that motivated these choices.

We simulate 100,000 NSBH binaries with parameters drawn from the following distribution. The black hole mass is chosen uniformly between 3 and 15 solar masses; the neutron star mass is chosen uniformly between 1 and 3 solar masses; the black hole dimensionless spin magnitude is chosen uniformly between 0 and 1 and the neutron star dimensionless spin magnitude is chosen uniformly between 0 and 0.05. The initial spin orientation for both bodies, the source orientation and the sky location are all chosen from an isotropic distribution.

Black holes observed in X-ray binaries can be used to estimate the BH mass distribution, though it is difficult to disentangle the individual masses and inclination

angle with only electromagnetic observations [**?**]. Using a population of ~ 20 low-mass X-ray binary systems with estimated masses, two separate works found that a BH mass distribution of $7.8 \pm 1.2 M_{\odot}$ fits the observed data well [**?**, **?**]. There is evidence that there is a "mass gap" between $3M_{\odot}$ and $5M_{\odot}$ where BHs will not form [**?**, **?**], although this may be due to observational bias [**?**]. When high-mass X-ray binary systems are considered the mass distribution increases to $9.2012 \pm 3M_{\odot}$, although a Gaussian model is a poor fit for these systems [**?**]. Evidence exists for a stellar mass black hole with mass $> 20M_{\odot}$ in the IC 10 X-1 x-ray binary [**?**, **?**]. We choose to use a uniform range of 3 to 15 solar masses for the black holes in our NSBH signal population. This is partly motivated by the considerations above, and partly by our concern of the validity of inspiral-only, point particle waveform models for high-mass NSBH systems. Observations of black hole spin have found spin values that span the minimum and maximum possible values for Kerr black holes [**?**], therefore we use a uniform black-hole spin distribution between 0 and 1.

Observations of NSs in binary systems other than NSBH binaries can be used to estimate the NS mass distribution. Using a population of 6 BNS systems with well constrained masses, Ozel et al. [**?**] found that the NS mass distribution was well fitted by $1.33 \pm 0.05 M_{\odot}$, in agreement with Kiziltan et al.'s result of $1.35 \pm 0.13 M_{\odot}$ [**?**]. However, non-recycled NSs in eclipsing high-mass binaries, as well as slow pulsars, are found to have a much wider mass distribution of $1.28 \pm 0.24 M_{\odot}$ [**?**]. Recycled NSs are found to have a higher range of masses, $1.48 \pm 0.2 M_{\odot}$, due to accretion [**?**]. However, it is expected that the black hole would form first in the vast majority of cases, which would remove the possibility of recycling. There is also evidence for a NS with a mass as high as $\sim 3M_{\odot}$ [**?**], which is very close to the theoretical upper limit on a NS mass of $\sim 3.2M_{\odot}$ [**?**]. While a conservative choice, we choose to use a uniform mass distribution between 1 and 3 solar masses for the NSs in our NSBH signal population.

The magnitude of the dimensionless spin, $\boldsymbol{\chi} = \boldsymbol{S}/m^2$, of a neutron star cannot be larger than ~ 0.7 [**?**] as the neutron star would break apart under the rotational force. However, it is rather unlikely that NS spins will have values as large as this in NSBH systems. At birth, neutron star spins are believed to be in the range 10 - 140 ms, corresponding to $\chi < 0.04$ [**?**, **?**]. Recycled neutron stars can have larger spin values [**?**], however they are unlikely to have periods less than 1 ms [**?**], corresponding

to a dimensionless spin of $\chi \sim 0.4$. The fastest spinning recycled neutron star observed in a BNS binary has a spin period of only 23 ms [**?**]. As astrophysical observations seem to suggest that large neutron spins will be unlikely in NSBH binaries we choose a uniform NS spin distribution between 0 and 0.05.

4.4 Waveform models

Matched-filter searches require an accurate model of compact binary mergers. In a companion work we investigate the agreement of different waveform families in the NSBH region of parameter space and find a considerable disagreement between waveforms produced by different waveform models, which will reduce detection efficiency [**?**].

In this work we wish to investigate the effects of spin, especially spin-induced precession, while understanding and mitigating any bias in our results due to the choice of waveform approximant. We therefore run all our simulations using two waveform approximants; TaylorT2 [**?**] and TaylorT4 [**?**].

PN waveforms, such as TaylorT2 and TaylorT4 are constructed by solving the PN equations of motion to obtain the binary orbits. It is assumed that the binary evolves adiabatically through a series of quasi-circular orbits. This is a reasonable assumption as it is expected that the emission of gravitational radiation will circularize the orbits of isolated binaries [**?**]. The equations of motion then reduce to series expansions of the center-of-mass energy $E(v)$ and the gravitational-wave flux $\mathcal{F}(v)$, which are expanded as a power series in the orbital velocity v:

$$E(v) = E_{\rm N} v^2 \left(1 + \sum_{n=2}^{6} E_i v^i\right), \tag{4.1}$$

$$\mathcal{F}(v) = F_{\rm N} v^{10} \left(1 + \sum_{n=2}^{7} \sum_{j=0}^{1} F_{i,j} v^i \log^j v\right). \tag{4.2}$$

The various coefficients (E_N,E_i,$\mathcal{F}_N$,$\mathcal{F}_i$) are reviewed in [**?**, **?**]. For terms involving the orbital contribution, the center-of-mass energy and gravitational wave flux are known to 3.5PN order ($n = 7$ in the parenthesis of 4.1) [**?**, **?**, **?**, **?**, **?**, **?**]. For terms involving the spin of the objects, the expansions of the energy and flux are complete to 2.5 PN order ($n = 5$ in the parenthesis of 4.1) [**?**, **?**, **?**]. In recent work, terms relating to the

coupling between the component spins and the orbit have also been computed to 3.5 PN order [**?**, **?**]. We choose not to use these terms in this work because terms relating to the spin(1)-spin(2), quadrupole-monopole and self-spin contributions are not yet known at 3 PN order, so we restrict the spin-related terms to 2.5 PN where these terms are fully known. We do not expect these terms to change the main conclusions of the work as these additional phase evolution terms will have little effect on the precessional evolution of a system.

The orbital phase, ϕ is then obtained via the energy balance equation

$$\frac{dE}{dt} = -\mathcal{F} \tag{4.3}$$

and by

$$\frac{d\phi}{dt} = \pi f. \tag{4.4}$$

Here the gravitational wave frequency f is given by twice the orbital phasing frequency, and is related to the orbital velocity by $v = (\pi M f)^{1/3}$ where M denotes the total mass of the binary.

The various approximants are constucted via *different* ways of obtaining the gravitational wave phase from the equations above.

4.4.1 TaylorT2 and TaylorF2

The TaylorT2 approximant is constructed by first calculating

$$B(v) = \left[\frac{E'(v)}{-\mathcal{F}(v)}\right]. \tag{4.5}$$

Here $[X]$ is used to indicate that X is calculated by first expanding it as a Taylor series. Then orbital terms larger than 3.5PN and spin terms larger than 2.5PN are discarded. This is because terms of this order would also depend on unknown terms in the expansion of the center-of-mass energy and the gravitational wave phase. As $B(v) = dt/dv$ the gravitational wave phase is therefore obtained according to

$$\phi(v) = \int \frac{v^3}{M} B(v) dv, \tag{4.6}$$

which can be integrated analytically. In the same manner $t(v)$ can be calculated according to

$$t(v) = \int B(v)dv. \tag{4.7}$$

$\phi(v)$ and $t(v)$ can then be numerically inverted to obtain $\phi(t)$ and $v(t)$, which are used to construct the waveform.

When constructing a TaylorT2 waveform, one begins at a fiducial starting frequency, chosen to be smaller than the lowest frequency over which to perform the matched-filter. In this work, we use 14Hz as the starting frequency. The waveform is terminated when the frequency reaches the MECO, which is the point where

$$\frac{dE(v)}{dv} = 0. \tag{4.8}$$

The TaylorF2 approximant is a frequency domain equivalent of the TaylorT2 approximant and is constructed using the stationary phase approximation [**?**, **?**, **?**, **?**]. The TaylorF2 waveforms can be expressed as an analytic expression of the form

$$\tilde{h}(f) = A(f; \mathcal{M}, D_L\theta_x)e^{-i\Psi(f;\lambda_i)}, \tag{4.9}$$

where $\tilde{h}(f)$ denotes the Fourier transform of $h(t)$, the time domain gravitational-wave strain, $\mathcal{M}$ denotes the chirp mass, D_L the luminosity distance to the source and θ_x describes the various orientation angles that only affect the amplitude and overall phase of the observed gravitational waveform [**?**]. The phase Ψ is given by

$$\Psi = 2\pi f t_c - \phi_c(\theta_x) + \sum_{i=0}^{7}\sum_{j=0}^{1} \lambda_{i,j} f^{(i-5)/3} \log^j f, \tag{4.10}$$

where t_c is the coalescence time and ϕ_c is a constant phase offset. The λ terms give the various coefficients of the orbital phase, which are summarized in [**?**, **?**]. TaylorF2 waveforms are usually terminated at the frequency corresponding to the inner-most stable circular orbit (ISCO) of a non-spinning system with the given masses [**?**].

4.4.2 TaylorT4 and TaylorR2F4

In contrast to the TaylorT2 approximant, the TaylorT4 approximant, introduced in [**?**] is formed by calculating

$$\frac{dv}{dt} = \left[\frac{-\mathcal{F}(v)}{E'(v)}\right] = A(v). \tag{4.11}$$

Similar to the TaylorT2 approximant, orbital terms larger than 3.5PN and spin terms larger than 2.5PN are discarded from $A(v)$. This is numerically solved to obtain $v(t)$ which can then be used to obtain the gravitational-wave phase. The TaylorT4 approximant uses the same start and termination conditions as the TaylorT2 approximant.

The TaylorR2F4 approximant, introduced in [**?**], is a frequency-domain analytical approximation of the TaylorT4 waveform model. It is constructed in the same manner as TaylorF2, however it uses

$$\frac{dt}{dv} = \left[\frac{1}{A(v)}\right] \tag{4.12}$$

instead of Eq. (4.5). In this case, while $A(v)$ is restricted as described above, $1/A(v)$ is truncated to a higher order in v. The additional "partial" terms that are obtained in the resulting PN expansion describe the difference between the TaylorT2 and TaylorT4 models. It has empirically been found that TaylorR2F4 matches best with TaylorT4 when $1/A(v)$ is expanded to 4.5PN order or 6PN order [**?**]. We only consider these two expansions of TaylorR2F4 in this work.

4.5 Method for assessing the performance of NSBH searches

In this section we describe the methods we use to assess the efficiency of template banks and the terminology that we will use in the rest of this work. The "overlap" between two waveforms h_1 and h_2 is defined as

$$\mathcal{O}(h_1, h_2) = (\hat{h}_1|\hat{h}_2) = \frac{(h_1|h_2)}{\sqrt{(h_1|h_1)(h_2|h_2)}}, \tag{4.13}$$

where (h_1, h_2) denotes the noise-weighted inner product

$$(h_1|h_2) = 4\,\mathrm{Re}\int_{f_{\min}}^{\infty} \frac{\tilde{h}_1(f)\tilde{h}_2^*(f)}{S_n(f)} df. \tag{4.14}$$

Here, $S_n(f)$ denotes the one sided power spectral density (PSD) of the noise in the interferometer. In this work, we model $S_n(f)$ with the aLIGO zero-detuned, high-power design sensitivity curve [**?**] and use a lower frequency cutoff, $f_{\rm min}$, of 15Hz.

As gravitational wave searches for binary mergers analytically maximize over an overall phase and time shift, we define the "match" between two waveforms to be the overlap maximized over a phase and time shift

$$\mathcal{M}(h_1, h_2) = \max_{\phi_c, t_c}(\hat{h}_1|\hat{h}_2(\phi_c, t_c)). \tag{4.15}$$

One can understand this match as the fraction of the optimal SNR that would be recovered if a template h_1 was used to search for a signal h_2.

We define the "fitting-factor" between a waveform h_s with unknown parameters and a bank of templates h_b to be the maximum match between h_s and all the waveforms in the template bank [**?**],

$$\mathrm{FF}(h_s) = \max_{h \in \{h_b\}} \mathcal{M}(h_s, h). \tag{4.16}$$

The "mismatch"

$$\mathrm{MM} = 1 - \mathrm{FF}(h_s) \tag{4.17}$$

describes the fraction of SNR that is lost due to the fact that the template in the bank that best matches h_s will not match it exactly due to the discreteness of the bank and due to any disagreement between the waveform families used to model the templates and the signals. In previous searches of LIGO and Virgo data using non-spinning template banks, the banks of signals were constructed so that the fitting factor would be greater than 0.97 for any non-spinning signal within the parameter space [**?**]. This was chosen as a balance between detection efficiency and computational cost. We also construct our aligned-spinning banks with this criterion.

When a set of fitting factors have been calculated one can quote an "average fitting factor" by taking the mean over all the values

$$\mathrm{FF}_{\rm av} = \langle \mathrm{FF} \rangle \,, \tag{4.18}$$

where $\langle X \rangle$ denotes the mean average of X. However, this measure can often be misleading. The aLIGO detectors have a direction-dependent and orientation-dependent sensitivity. Systems that are poorly aligned with respect to the detector may not have sufficient SNR to be detected, regardless of the fitting factor. To account for this we make use of the "effective fitting factor", first defined in [**?**] as

$$\mathrm{FF}_{\mathrm{eff}} = \left(\frac{\left\langle \mathrm{FF}^3 \sigma_i^3 \right\rangle}{\left\langle \sigma_i^3 \right\rangle} \right)^{1/3}. \tag{4.19}$$

Here $\sigma_i = \sqrt{(h_i|h_i)}$, which describes the optimal SNR of h_i. The cube of the effective fitting factor gives, above an arbitrary SNR threshold, the ratio between the fraction of NSBH signals that would be recovered with the discrete template bank that was used and a theoretical continuous template bank that would recover 100% of signal power for *any* NSBH waveform. We therefore define the "signal recovery fraction" as $\mathrm{FF}_{\mathrm{eff}}^3$.

4.6 A new algorithm for constructing template banks of aligned-spin NSBH waveforms

In [**?**] we proposed a method for generating a geometrically-placed bank of aligned-spin systems that can be used to search for BNS systems in the advanced detector era. In this section we adapt the methods presented in that work to the case of NSBH systems and describe how to generate template banks that can recover aligned-spin NSBH waveforms. These banks are applicable for waveforms modelled using either the TaylorT2 approximant or the TaylorT4 approximant.

A bank of templates should be placed such that any putative signal within the parameter space of interest would be recovered with a loss in SNR that is always less than some predefined value, usually taken to be 3% [**?**, **?**, **?**, **?**, **?**, **?**]. To determine the maximum spacing between templates that meets this criterion, the parameter space metric is used. This approximates the distance between any two points that are close in the parameter space [**?**]

$$\mathcal{O}(h(\boldsymbol{\theta}), h(\boldsymbol{\theta} + \delta\boldsymbol{\theta})) = 1 - \sum_{ij} g_{ij}(\boldsymbol{\theta})\, \delta\theta^i\, \delta\theta^j, \tag{4.20}$$

with the metric given by

$$g_{ij}(\boldsymbol{\theta}) = -\frac{1}{2}\frac{\partial^2 \mathcal{O}}{\partial\delta\theta^i \partial\delta\theta^j} = \left(\frac{\partial h(\boldsymbol{\theta})}{\partial\theta^i}\middle|\frac{\partial h(\boldsymbol{\theta})}{\partial\theta^j}\right). \tag{4.21}$$

Here $\boldsymbol{\theta}$ describes the parameters of the signal, in this case the masses and the spins. This is also commonly referred to as the Fisher metric.

Obtaining an analytic solution for Eq. 4.21 is much simpler in the frequency domain and therefore frequency-domain waveform models are commonly used when placing a bank of templates [**?**, **?**, **?**]. We follow that approach and consider only frequency-domain metrics here. It is important to carefully consider which coordinates to use as parameters when using this metric as an approximation to the parameter space distance. If one were to naively use the masses and spins directly as coordinates it would result in a parameter space metric with a large amount of extrinsic curvature, and Eq. (4.20) would only be valid for small ranges of $\delta\theta^i$. In previous searches for non-spinning systems, the "chirp times" were used [**?**], defined as,

$$\tau_0 = \frac{5}{128}(\pi M)^{-5/3}\eta^{-1} \tag{4.22a}$$

$$\tau_3 = \frac{\pi}{4}(\pi M)^{-2/3}\eta^{-1}, \tag{4.22b}$$

as these are the two combinations of the masses that minimize extrinsic curvature.

When the template waveforms include spin it is difficult to identify a parameterization of the waveform for which the metric is locally flat. Instead, in [**?**] we constructed a metric that uses the various coefficients of the expansion of the orbital phase, given by the various λ_i terms in Eq. (4.10), directly as coordinates. Using these coordinates, the parameter space is globally flat. However, for the TaylorF2 metric including terms up to 3.5PN order, the parameter space is 8-dimensional. The physical sub-space forms a 4-dimensional manifold within this parameter space.

To deal with the increased dimensionality of the space we perform two coordinate transformations [**?**, **?**]. These two coordinate transformations map points from the λ_i coordinates into a Cartesian coordinate system where the principal directions are mapped using coordinates denoted by ξ_i. Specifically, the first coordinate transformation uses the eigenvectors and eigenvalues of the λ_i metric to transform to a Cartesian

coordinate system. A Principal Component Analysis is then performed to rotate into the frame given by the principal directions of the parameter space. In this Cartesian coordinate system of principal directions we can assess the "effective dimension" of the parameter space; i.e., the number of directions in which templates actually need to be placed in order to achieve the desired coverage. For the case of the BNS parameter space with the aLIGO PSD we found that many of the directions had an extremely small extent and could be neglected entirely. We found that a 2-dimensional lattice could efficiently cover the entire space of aligned-spin BNS waveforms [**?**].

Our geometrical placement method is not specific to the BNS area of the parameter space. However, some modifications to the method were necessary when placing a template bank of NSBH waveforms. Our BNS aligned-spin template bank, as described in [**?**], was given in terms of the positions of the points in the 8-dimensional Euclidean parameter space, ξ_i. These points do not correspond directly to physical masses and spins. For this study we want to use time domain template families and therefore we must translate the bank into physical parameters. However, if a set of ξ_i values is given it will, in general, not be possible to find a set of masses and spins that give the exact ξ_i values. As templates are normally placed in a 2-dimensional lattice, we need only to find a physical point that has the corresponding values of ξ_1 and ξ_2 and *any* value of the other ξ_i values that correspond to a waveform within the physically allowed manifold. For some cases where a 2-dimensional lattice is not sufficient to cover the space we will also specify values of ξ_3 and ξ_4. We attempt to find a physical solution that is sufficiently close to the desired point using a numerical solution. We generate a large set of points in the mass and spin space and map these points to the ξ_i parameters. For each template we then find the closest point from our large set of physical points. We then proceed to iteratively test physical points in the vicinity to find a match of at least 0.9999 with the intended position. If the template is within the physically allowed parameter space we can generally find a physical point that has the desired match with the intended ξ_i point. Templates on the boundaries of the space, might have an overlap as low as ~ 0.97 with the edge of the physical parameter space. Our method pushes such points back into the desired physical space thereby providing a slight *improvement* in the bank coverage. This method also provides an easy method to determine the extent of the physical space: if *no* physical point is found with 0.97 or higher match with the ξ_i position then that

point is not within the physical extent of the parameter space and no template needs to be placed there.

The downside to our brute-force numerical method is that it is currently not computationally efficient; generating a bank with this numerical technique can take $O(10)$ hours when running on ~ 500 CPU cores. The cost of placing a bank using this method, however, is negligible when compared to the cost of filtering data against a bank of templates if a single bank is used to filter O(days) of data. If the bank is regenerated every hour, as in previous searches of LIGO and Virgo data [**?**], this cost would not be negligible. We note that it should be possible to optimize our implementation to obtain a significant speed increase over what we quote above.

The TaylorF2 metric can be used to place a bank of waveforms modelled with the TaylorT2 approximant. However, we also require that our template placement algorithm place a bank of waveforms that can detect aligned-spin signals modelled using TaylorT4 with no more loss in SNR than that specified by the minimal match of the bank. This will allow us to investigate the efficiency of aligned-spin banks to search for precessing NSBH signals using two waveform models. Using two models will help to mitigate any bias in our results that arises due to the choice of waveform approximant. We investigate the distribution of fitting factors when using a template bank constructed using the TaylorF2 metric to search for aligned-spin TaylorT4 NSBH signals in section 4.8 and find that this would result in a reduction of sensitivity. We therefore make use of a metric that models the TaylorT4 waveform well. To do this we use the TaylorR2F4 waveform model. We have found that restricting the TaylorR2F4 model to terms no larger than 4.5PN and placing a bank of templates using the ensuing metric is sufficient to cover the TaylorT4 parameter space. This is a 12-dimensional metric. We then perform the same rotations as for the TaylorF2 metric to identify the ξ_i directions for our TaylorR2F4 parameter space and proceed in the same manner as described above.

In contrast to BNS mergers, NSBH systems can merge in the sensitive band of the advanced detectors. Existing non-spinning template placement algorithms [**?**, **?**, **?**, **?**, **?**], as well as our aligned-spin algorithm must use the same termination frequency when modelling waveforms across the parameter space. The standard approach is to assume that the waveforms will follow the TaylorF2, or TaylorR2F4, evolution up to the Nyquist frequency, usually 2048Hz. For BNS systems, the merger generally occurs

Template bank	Approximant	Waveform cut
Geometric non-spinning bank	TaylorF2	1000
Geometric non-spinning bank	TaylorR2F4 (up to 4.5PN)	1000
Geometric aligned-spin bank	TaylorF2	1000
Geometric aligned-spin bank	TaylorF2	400
Geometric aligned-spin bank	TaylorF2	240
Stochastic aligned-spin bank	TaylorF2	Dyna
Geometric aligned-spin bank	TaylorR2F4 (up to 4.5PN)	1000
Geometric aligned-spin bank	TaylorR2F4 (up to 4.5PN)	400
Geometric aligned-spin bank	TaylorR2F4 (up to 4.5PN)	240
Stochastic aligned-spin bank	TaylorR2F4 (up to 4.5PN)	Dyna

Table 1: The sizes of the various template banks that are used in this work. All of these banks ar
$\in [1, 3)M_\odot$; BH dimensionless spin $\in [-1, 1]$; NS dimensionless spin $\in [-0.05, 0.05]$. For all banks t
frequency cutoff of 15Hz.

above 1000Hz where the sensitivity of gravitational wave interferometers falls off and therefore little power is incurred between 1000Hz and Nyquist. Even a $(3 + 3)M_\odot$ BNS has an ISCO with a frequency of 730Hz. In contrast, a $(15+3)M_\odot$ NSBH system has an ISCO frequency at 240Hz. We must therefore consider what frequency cutoff is most appropriate to use when placing a bank of NSBH waveforms.

We found that using an upper frequency cutoff that is higher than the waveform's termination frequency results in overcoverage in the parameter space. This result is expected as the sub-dominant PN terms can have a significant effect in the late part of the evolution, causing systems with the same chirp masses but different spins and mass ratio to diverge faster. Therefore we use an upper frequency cutoff of 1000Hz for all waveforms within the NSBH parameter space to generate a template bank that will cover to the desired minimal match. However, as this template bank will overcover at least the high mass end of the parameter space we also investigate the efficiency of banks placed with smaller upper frequency cutoffs in section 4.8.1. This choice will be an important consideration in the advanced detector era given limits on computational power for conducting NSBH searches.

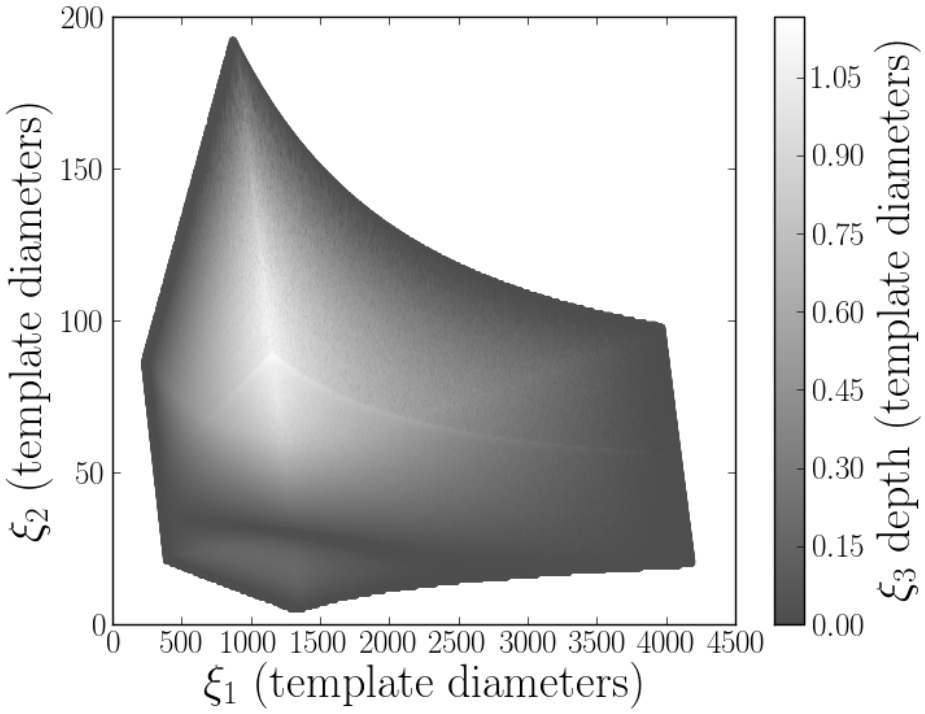

Figure 28: The depth of the physically possible range of ξ_3 (left) and ξ_4 (right) values as a function
coordinates have been scaled such that one unit corresponds to the coverage diameter of a templ
advanced LIGO sensitivity curve with a 15Hz lower frequency cut off

4.7 Constructing template banks of aligned-spin NSBH waveforms with our new algorithm

We begin by creating a template bank using the TaylorF2 parameter space metric. We first explore the space to assess the effective dimensionality and to determine whether the 2-dimensional placement used to cover the BNS space in [**?**] is applicable to the NSBH space. We do this by creating a set of 10^7 points drawn uniformly from the chosen range of NSBH masses and spins. We then transform these points into the ξ_i coordinates. In Fig. 28 we show the extent of the dominant two directions (ξ_1 and ξ_2). The color shows, respectively, the depth of the third direction (ξ_3) and the fourth direction (ξ_4). The fifth and subsequent directions are, as in the BNS space, small enough to be ignored completely.

From these plots we can see that the extent of the space in all but the ξ_1 and ξ_2 directions is small in most regions. In these areas a 2-dimensional lattice of template points would suffice to cover the parameter space. However, there is a small region in the center of the parameter space where the depth of the third direction is not negligible. Therefore, to cover this space we follow [**?**] and initially place a 2-dimensional lattice in the ξ_1, ξ_2 coordinates. Then, where necessary, templates are stacked in the ξ_3 direction. The density of this stacking is chosen such that the loss in match due to the depth of the third direction can never be larger than 0.01. As the 2-dimensional lattice is placed to ensure that matches will not be less than 0.97 in a 2-dimensional plane, and as each direction in our Euclidean parameter space is orthogonal, there are therefore regions of the parameter space where the fitting factor can be as low as 0.96. However, these regions are small and the mean fitting factor, as we will show, is still much larger than 0.97. This bank, constructed using the TaylorF2 parameter space metric, contains 801,183 templates, of which 134,807 were added by the stacking process. For ease of comparison Table 1 gives the sizes and properties of all the banks that are used in this work.

We next construct a bank of template waveforms using the TaylorR2F4 parameter space metric. We begin by exploring the parameter space to assess the effective dimensionality. In Fig. 29 we show the depths of the ξ_3 and ξ_4 directions as a function of ξ_1 and ξ_2 for the TaylorR2F4 parameter space. We immediately notice that the degeneracies present in the TaylorF2 space, which allow us to use a 2-dimensional

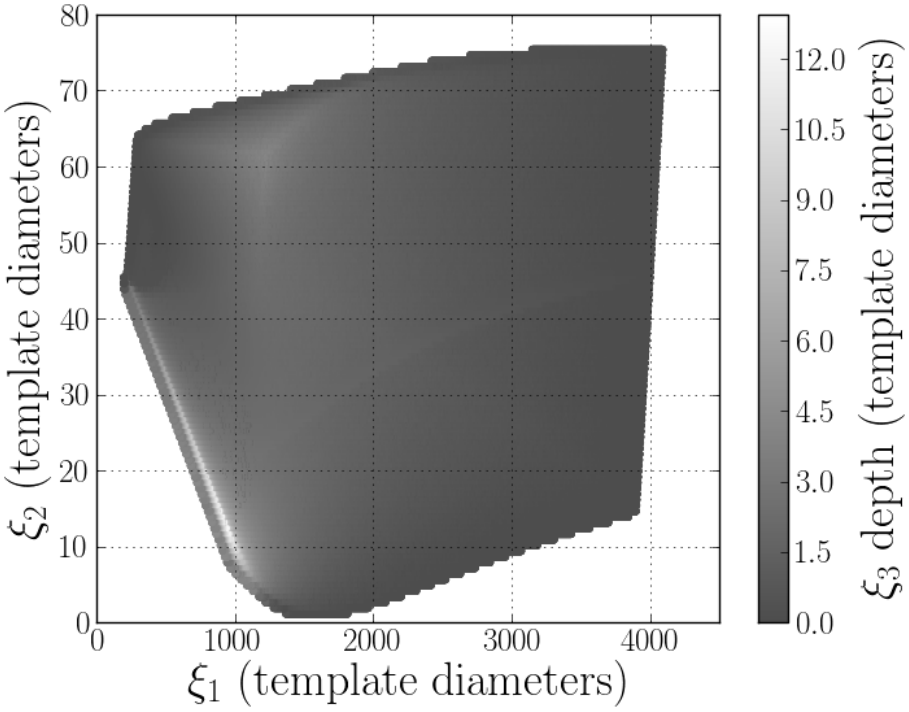

Figure 29: The depth of the physically possible range of ξ_3 (left) and ξ_4 (right) values as a function
ξ_i coordinates have been scaled such that one unit corresponds to the coverage diameter of a temp
advanced LIGO sensitivity curve with a 15Hz lower frequency cut off

placement, are much weaker in the TaylorR2F4 parameter space. For this space there is substantial depth in the third direction. In one small region it is wider than 10 template diameters. The median depth in this direction, however, is only one template diameter.

If the depth in the third direction was larger in all regions, the most efficient placement scheme would be to place a template bank in a 3-dimensional $A_n^\star$ lattice [**?**]. However, in regions where the depth of the 3rd direction is small, the 3-dimensional lattice, when flattened into the 2-dimensional space, would cause an overcoverage. We therefore tried both a 3-dimensional lattice placement and a 2-dimensional placement, followed by stacking in the 3rd direction as we used for the TaylorF2 bank. Additionally, unlike in the TaylorF2 space, the depth of the fourth dimension is not negligible. However, as in most places the width in that direction is small, the stacking technique can also be used to cover the depth of the 4th dimension when needed.

When we choose to employ a 3-dimensional lattice we find that 1,805,036 templates are needed to cover the space, 90,463 of which were added due to stacking in the 4th direction. In contrast, when we use a hexagonal lattice followed by stacking in both the 3rd and 4th directions we find that 1,100,277 templates are needed, of which 741,626 were added by the stacking process. It may seem surprising that the 2-d hexagonal lattice requires less templates than the 3-d $A_n^\star$ lattice. In fact, it would still require less templates even if the depth of the third direction was large in all regions of the space. The reason for this is that the $A_n^\star$ placement *guarantees* that all points within the 3-dimensional space will have a fitting factor of at least 0.97. With the hexagonal placement followed by stacking, there are points in the space where the fitting factor can be as low as 0.96 (when the depth of the 4th dimension is significant this can be as low as 0.95). If we were to require that all points within the space *must* have a fitting factor of at least 0.97, our hexagonal lattice would need to be placed to a minimal match of 0.98. For comparison, we generated a 3-dimensional lattice with a minimal-match of 0.96, this bank contained 1,175,523 templates. The 3-dimensional lattice is still less efficient than the 2-dimensional lattice. This can be attributed, as described above, to the fact that the depth of the 3rd direction is not large in all areas of the parameter space. In some areas a 2-dimensional lattice, without any stacking, is sufficient to cover the parameter space. An alternative approach might be to use a 3-dimensional lattice of points only in regions where it is needed and a

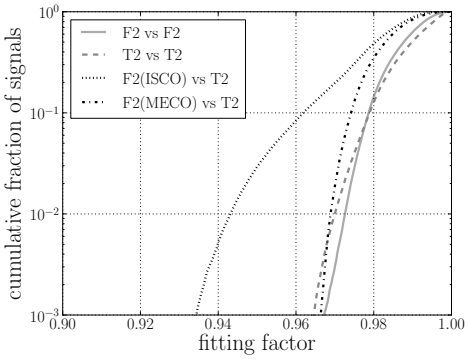

Figure 30: Fitting factor between a set of aligned-spin NSBH signals and our geometrically placed aligned-spin template bank placed using the TaylorF2 metric. Shown when both templates and signals are generated using the TaylorF2 approximant (gray solid line) and when both are modelled with TaylorT2 (gray dashed line). Also shown when the signals are modelled with TaylorT2 and the templates modelled with TaylorF2 waveforms terminated at ISCO (black dotted line) and TaylorF2 waveforms terminated at MECO (black dot-dashed line). Results obtained using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

2-dimensional lattice elsewhere, we did not investigate that here. For the simulations in the following sections, we use the hexagonal lattice with stacking as the method for placing banks of templates for the TaylorT4 approximant.

4.8 Results I: Validating the new template bank placement for aligned-spin systems

In this section we demonstrate that our aligned-spin template banks acheive the level of coverage they are constructed for when used to search for aligned-spin signals. We also compare our banks to banks generated using a stochastic placement algorithm [**?**, **?**, **?**, **?**] and show that our method acheives the same level of coverage with fewer templates.

To verify the performace of our aligned-spin template banks we compute the fitting factors between the banks and a set of 100,000 aligned-spin NSBH waveforms. These waveforms are drawn from the distribution that we describe in section 4.3, except that the spins are all aligned (or anti-aligned) with the orbital angular momentum.

In Fig. 30 we show the results of this test using the template bank constructed

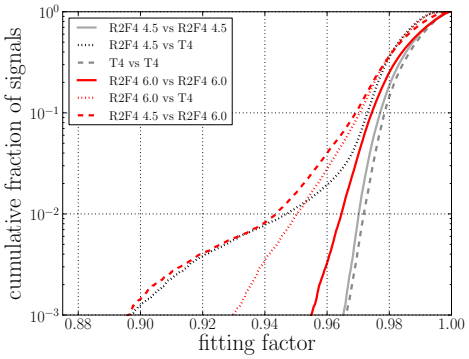

Figure 31: Fitting factor between a set of aligned-spin NSBH signals and our geometrically placed aligned-spin template bank placed using the TaylorR2F4 metric. Shown are comparisons between TaylorT4 waveforms, TaylorR2F4 waveforms including terms to 4.5PN order and TaylorR2F4 waveforms including terms to 6PN order. Results obtained using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

with the TaylorF2 metric. We show results when both template waveforms and signals are modelled using the TaylorF2 approximant, when both are modelled using the TaylorT2 approximant and when we model the template waveforms with TaylorF2 and the signals with TaylorT2. In both cases where the same waveform model was used almost all of the fitting factors were greater than 0.97. The bank generation was successful.

The lowest matches in the TaylorF2 vs TaylorF2 results were in cases where a system with low mass ratio was recovered with a template with a high mass ratio, or vice-versa. These are systems where the degeneracy between the spins and the mass ratio [**?**] causes the phase evolution of the two systems to be very similar and therefore the match predicted by the metric is higher than 0.97. However, the system with the larger black hole mass will terminate at a significantly lower frequency than the system with the smaller black hole mass and some power is lost due to the difference in termination frequencies, which is not predicted by the metric.

The difference in termination conditions is also the reason why we see comparitively poorer performance when using TaylorF2 waveforms, terminated at the ISCO frequency, to search for TaylorT2 signals. The TaylorT2 signals terminate when the evolution becomes unphysical, either at the MECO or where the frequency spuriously begins to drop. In some cases, especially when the spins are large, these can

correspond to rather different termination frequencies. To demonstrate this we also show the performance of searching for TaylorT2 signals with TaylorF2 waveforms, but where we terminate the TaylorF2 waveforms using the same cut-off frequency that TaylorT2 waveforms would have at the given masses and spins. This gives a much more comparable performance to the TaylorF2 vs TaylorF2 and TaylorT2 vs TaylorT2 cases.

In Fig. 31 we repeat this test using the template bank constructed with the TaylorR2F4 metric, with terms restricted to 4.5PN order. We show results when the template waveforms and signals are modelled with varying approximants. We use TaylorR2F4 with terms up to 4.5PN order, TaylorR2F4 with terms up to 6PN order and TaylorT4. We can see from this figure that using TaylorR2F4 template waveforms with terms only to 4.5PN order would not be satisfactory when conducting searches for signals modelled with the TaylorT4 approximant. However, we note that when this bank is used with either TaylorT4 templates or TaylorR2F4 templates including terms up to 6PN order the coverage is much better. When TaylorT4 is used to model both the signals and the template waveforms we find that $> 99\%$ of the fitting factors are greater than 0.97. In this plot the TaylorR2F4 waveforms are terminated at the same frequency (the MECO frequency) as the TaylorT4 waveforms.

The TaylorR2F4 metric, with terms up to 4.5PN, is sufficient to place a bank of templates to cover waveforms modelled by the TaylorT4 approximant. However, when performing the matched-filtering the templates must be modelled with either TaylorT4 or TaylorR2F4 with terms up to 6PN order.

In Fig. 32 we also show the performance of a bank placed using the TaylorF2 metric to search for TaylorT4 aligned-spin signals. We assess the performance when the templates are modelled using TaylorF2, TaylorT2 and TaylorT4 approximants. Even when TaylorT4 is used to model both template waveforms and signals, 10% of signals are recovered with fitting factors smaller than 0.95. The TaylorF2 metric does not acheive the desired coverage for TaylorT4 waveforms. In a companion work we investigate how the disagreement of different waveform families in the NSBH region of parameter space will reduce detection efficiency [**?**].

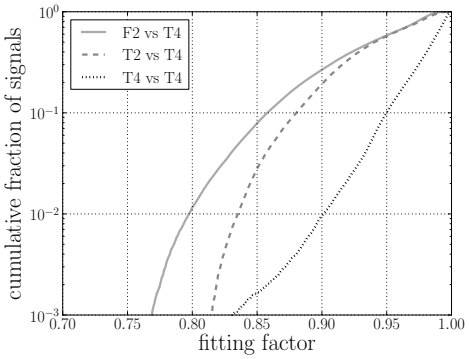

Figure 32: Fitting factor between a set of aligned-spin NSBH signals modelled with the TaylorT4 approximant and our template bank of aligned-spin signals placed using the TaylorF2 parameter space metric. Shown are the fitting factors when the templates used are modelled using the TaylorF2 approximant (grey solid line), TaylorT2 (grey dashed line) and TaylorT4 (black dotted line). Results obtained using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

4.8.1 Varying the upper frequency cutoff and comparison with stochastic placement algorithms

Filtering $\sim 10^6$ templates against data from advanced gravitational-wave detectors will require a large amount of computing power. It would therefore be desireable if we could reduce the overcoverage that is incurred in the high mass region of the parameter space when using an upper frequency cutoff of 1000 Hz. An alternative "stochastic" placement scheme, based on randomly picking points in the space and only retaining points which are not close to points already in the bank [**?**, **?**, **?**], is capable of using an upper frequency cutoff that varies with mass [**?**]. However, this method is known to pack templates more densely than a geometrical lattice [**?**]. We found that using a stochastic method to cover this NSBH space with the same covering criterion required 971,105 (1,327,175) templates when using the TaylorF2 (TaylorR2F4) metric to place the bank. In both cases this is $\sim$ 20% larger than our geometric algorithm using a constant upper frequency cutoff of 1000 Hz. It is also possible to generate the geometric bank with a lower upper frequency cutoff. This will require less templates, but will not reach the desired coverage in the lower mass regions of the parameter space. In Fig. 33 we compare the efficiency of geometric banks placed using a 240Hz, 1000Hz and 400Hz upper frequency cutoff. These correspond to roughly the lowest

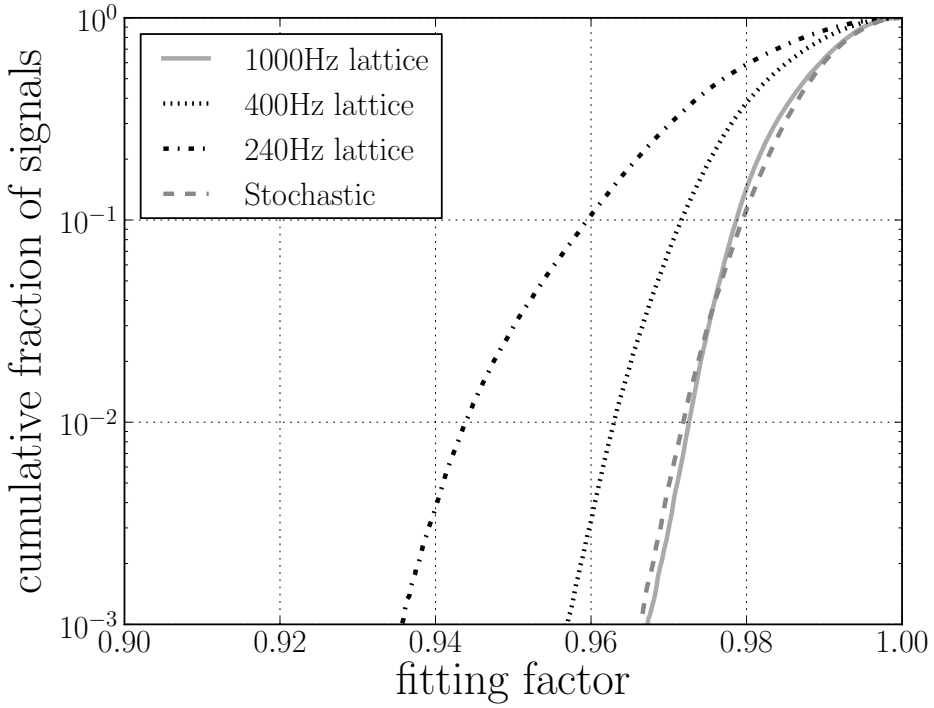

Figure 33: Fitting factor between a set of aligned-spin NSBH signals and a template bank of alig
used in the construction metric. Shown for template banks placed using the TaylorF2 metric a
approximant (left). Also shown for template banks placed using the TaylorR2F4 metric and w
approximant (right). The performance of using a stochastically placed template bank with varyin
zero-detuned, high-power advanced LIGO sensitivity curve wit

possible ISCO frequency, the highest and an "average" system. The sizes of these banks are shown in Table 1. As expected we notice a number of systems recovered with fitting factors less than 0.97 when the upper frequency cutoff is reduced. We also compare with the performance of a stochastic placement algorithm, which uses a varying upper frequency cutoff. The performance of the stochastic bank is very comparable to the 1000Hz bank when using the TaylorF2 metric. When using the TaylorR2F4 metric the stochastic bank, which was placed using 10^9 seed points, seems to be struggling to achieve the necessary coverage in certain regions of the space. As the stochastic placement algorithm only uses a finite number of sample points, it is known that it can leave holes in the parameter space, resulting in undercoverage [**?**].

We plan to adapt the geometric placement algorithm to allow the upper frequency cutoff to vary over the space, however we leave this investigation for future work. We note that the minimal match and *lower* frequency cutoff of the bank can also be modified to reduce the number of templates and balance the computational cost [**?**].

4.9 Results II: Template bank performance when searching for generic NSBH signals

In this section we evaluate the efficiency of searching for generic NSBH systems using template banks of non-spinning waveforms. Template banks of non-spinning waveforms were used to search for NSBH signals in data from LIGO and Virgo's most recent science runs [**?**, **?**, **?**, **?**]. We demonstrate that ignoring the effects of spin when conducting searches for NSBH systems in the advanced detector era will significantly decrease the rate of NSBH observations and impose a selection bias against systems with large spins and large m_{BH}/m_{NS}. We then evaluate the efficiency of searching for generic NSBH systems using our new template bank of aligned-spin waveforms. We calculate the improvement gained by using our new bank when compared to a non-spinning bank.

4.9.1 Performance of non-spinning template banks when searching for generic NSBH signals

We compute fitting factors between a set of 100,000 generic, precessing NSBH signals and a bank of non-spinning template waveforms. The precessing signals are drawn

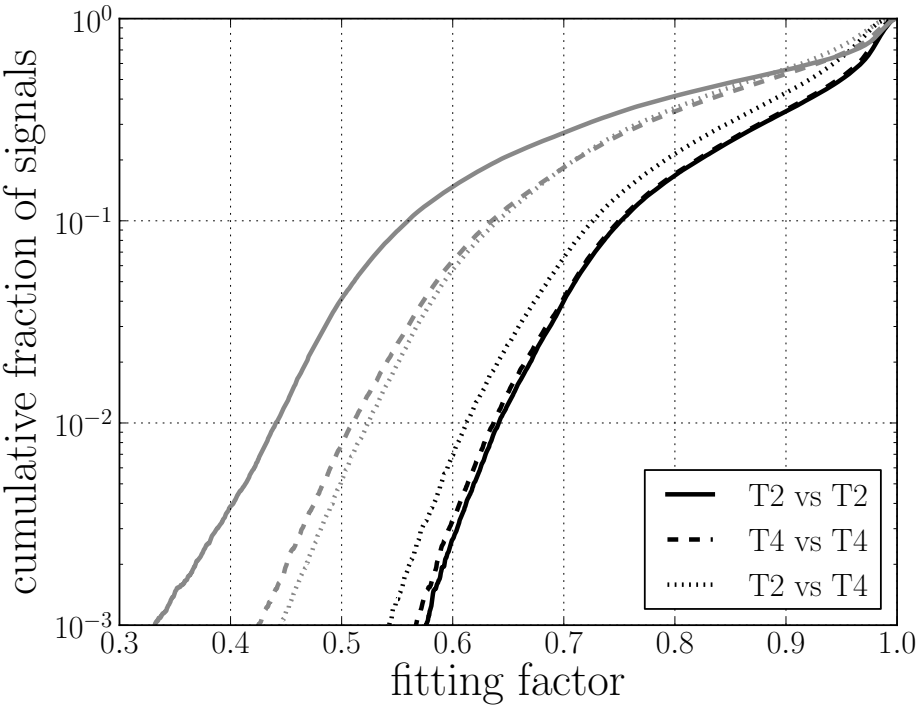

Figure 34: Fitting factor between a set of generic, precessing, NSBH signals and a template bank are generated using the TaylorT2 approximant (black solid line) and the TaylorT4 approximant (bl TaylorT2 and the signals are modelled using TaylorT4 (black dotted line). For comparison the sam plotted in grey. Plotted over the full range of fitting factors (left) and zoomed in to show only fitti signals are drawn from is described in section 4.3. The template bank construction is described i advanced LIGO sensitivity curve with a 15Hz lowe

from the distribution that we describe in section 4.3. To mitigate any bias that arises due to the choice of waveform approximant we run the simulation twice. First we use the TaylorT2 approximant for both signal and template waveforms and a template bank designed to obtain a fitting factor of at least 0.97 for any TaylorT2 non-spinning signal. The simulation was then repeated using the TaylorT4 approximant for both signal and template waveforms and a bank designed with the same fitting factor criterion for TaylorT4 signals. These banks were constructed using the methods described to create aligned-spin banks in section 4.7 but with the spins set to 0.

The results of this simulation can be seen in Fig. 34. From this we can calculate the mean and median values of the fitting factor over the signal distribution that we used. The mean fitting factor of the signals is 0.82 (0.84) for the TaylorT2 (TaylorT4)

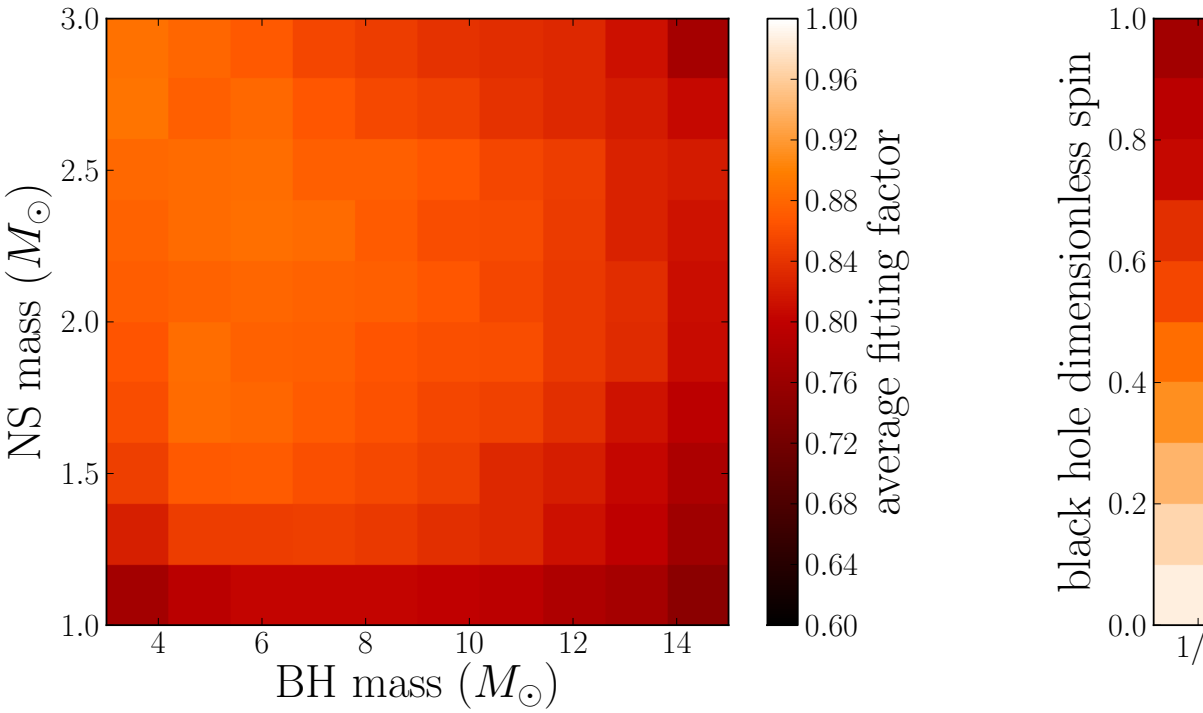

Figure 35: Average fitting factor between a set of generic, precessing, NSBH signals and a templa masses (left) and as a function of the mass ratio and the black hole dimensionless spin magnitude using the TaylorT4 approximant. The distribution that the NSBH signals are drawn from is de high-power advanced LIGO sensitivity curve with a 15H

approximant, while the median fitting factor was 0.86 (0.88). In both cases the distributions have long tails, with some systems recovered with less than 30% of their optimal SNR. We also show results where we have modelled the templates using the TaylorT2 approximant and the signals using the TaylorT4 approximant. In this case the mean fitting factor is 0.84 and the median is 0.87. We notice that fewer signals are recovered with high fitting factors (> 0.95) than in the other two cases, but we notice that at lower values of fitting factor the performance is very similar to the TaylorT4 vs TaylorT4 case. The slight *improvement* of the TaylorT2 vs TaylorT4 case at lower fitting factors can be attributed to the fact that the TaylorT2 bank is $\sim 20\%$ larger than the TaylorT4 bank and therefore has more freedom to match TaylorT4-modelled spinning signals.

In Fig. 35, we show the mean fitting factor as a function of the intrinsic parameters of the system when both templates and signals were modelled with the TaylorT4

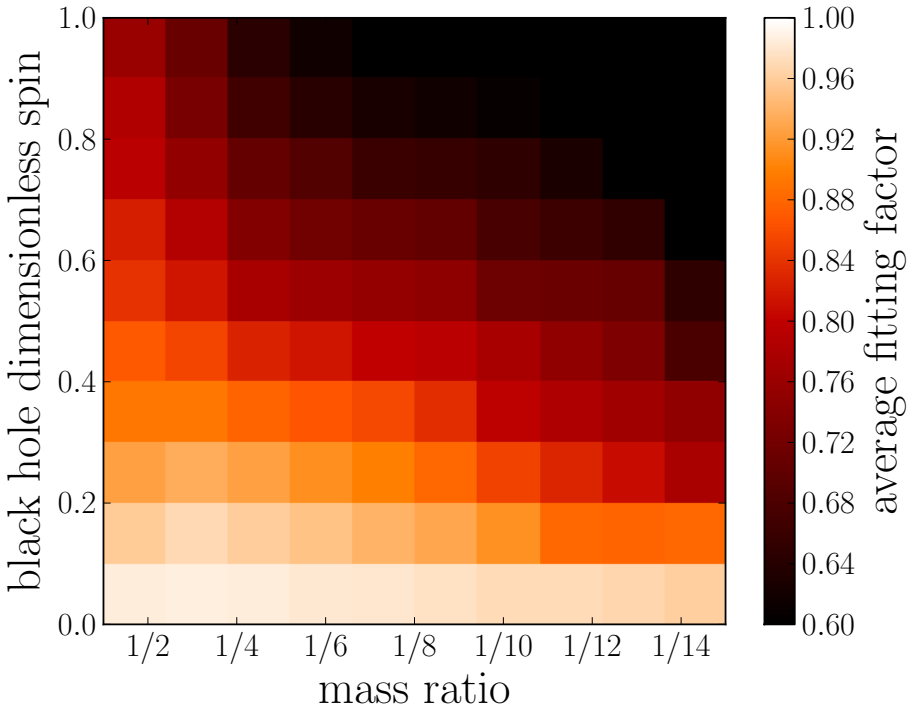

Figure 36: Average fitting factor between a set of generic, precessing, NSBH signals and a template the black hole dimensionless spin magnitude (right). Shown when both the template waveforms ar waveforms are modelled with TaylorT2 and the signals are modelled with TaylorT4 (right). The res panel of Fig. 35. The distribution that the NSBH signals are drawn from is described in section 4 LIGO sensitivity curve with a 15Hz lower fre

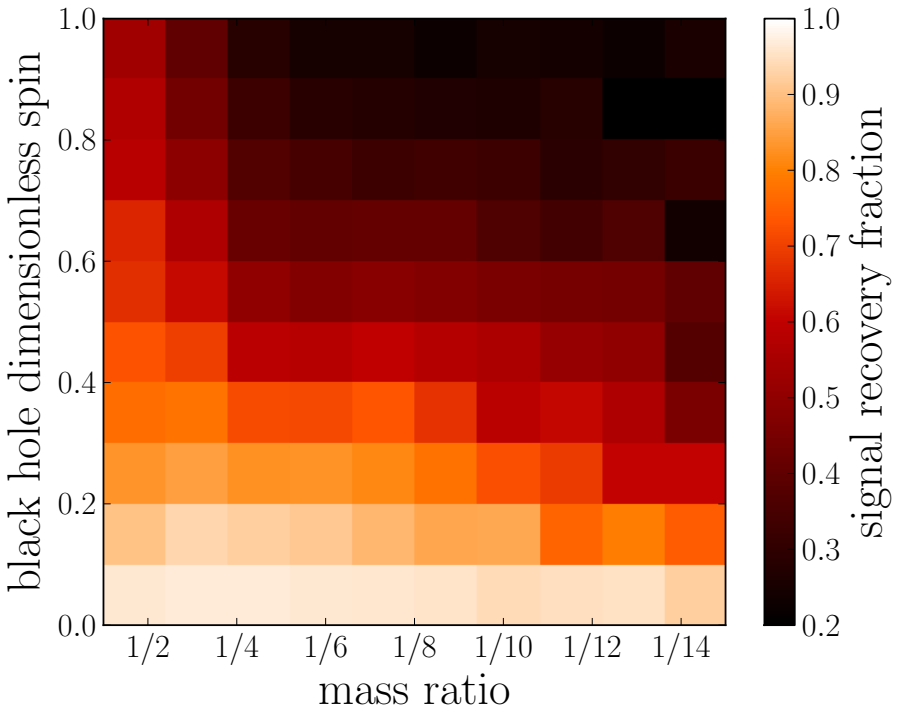

Figure 37: The signal recovery fraction obtained for a set of generic, precessing, NSBH signals and ratio and the black hole dimensionless spin. Shown when both the template waveforms and the sig waveforms and the signals are modelled with TaylorT4 (right). The distribution of the signal recov average fitting factors shown in Figs. 35 and 36. The distribution that the NSBH signals are dr zero-detuned, high-power advanced LIGO sensitivity curve with

approximant. For comparison, in Fig. 36 we show the mean fitting factor as a function of the spin magnitude and mass ratio for the TaylorT2 vs TaylorT2 results and the TaylorT2 vs TaylorT4 results. In both cases the results are similar to the TaylorT4 vs TaylorT4 case, which indicates that the results are not suffering from a significant bias due to the choice of waveform approximant. However, we note that when using TaylorT2 as the signal model, the performance of the non-spinning banks is worse for high spin, unequal mass systems than when using TaylorT4 as the signal model.

In Fig. 37 we show the signal recovery fraction as a function of the BH spin magnitude and the mass ratio. The signal recovery fraction is defined in section 4.5. It is clear that using a non-spinning bank to search for NSBH systems will result in a considerable reduction in the NSBH detection rate. In addition, the ability to detect systems with high spin, especially systems that also have unequal masses, is

especially poor. We note that these efficiencies would be improved by using non-spinning templates outside of the chosen mass ranges, for example BNS or binary black-hole template waveforms, or even templates with unphysical mass parameters [?, ?].

4.9.2 Performance of aligned-spin template banks when searching for generic NSBH signals

With the template banks of aligned-spin systems described in section 4.7, we are able to recover aligned-spin systems modelled with either the TaylorT2 or TaylorT4 approximant with fitting factors greater than 0.97 in $> 99\%$ of cases, as shown in section 4.8. If we use these banks to search for precessing systems modelled with the same approximants, any loss in signal power, beyond that lost due to the spacing of the aligned-spin bank, is entirely due to precession. We now assess the performance of these aligned-spin banks when searching for generic, precessing NSBH signals and identify regions of the parameter space where precessional effects cause a significant loss in detection rate.

Our signal population is a set of 100,000 precessing NSBH signals. This distribution was described in section 4.3. For comparison this is the *same* set of signals as we used in section 4.9.1. As before, we will assess fitting factors using both the TaylorT2 and TaylorT4 models to mitigate any bias arising from choice of waveform model. When TaylorT2 is used as the signal model, we will use the bank of aligned-spin systems that was placed using the TaylorF2 metric and a 1000Hz upper frequency cutoff and model the templates using the TaylorT2 approximant. When TaylorT4 is used as the signal model, we will use the bank of aligned-spin systems placed using the TaylorR2F4 metric and model the templates with TaylorT4. The placement of these banks was described in section 4.7.

The results of these simulations can be seen in Fig. 34, where we also compare with the results obtained in section 4.9.1 when using non-spinning template banks. We can clearly see from Fig. 34 that the distribution of fitting factors for the case when both signals and templates were modelled with TaylorT2 agrees well with the case when both were modelled with TaylorT4. This indicates that we have disentangled precessional effects from waveform-dependent effects and our results are free of any bias due to the choice of waveform model. The mismatches seen here, beyond that

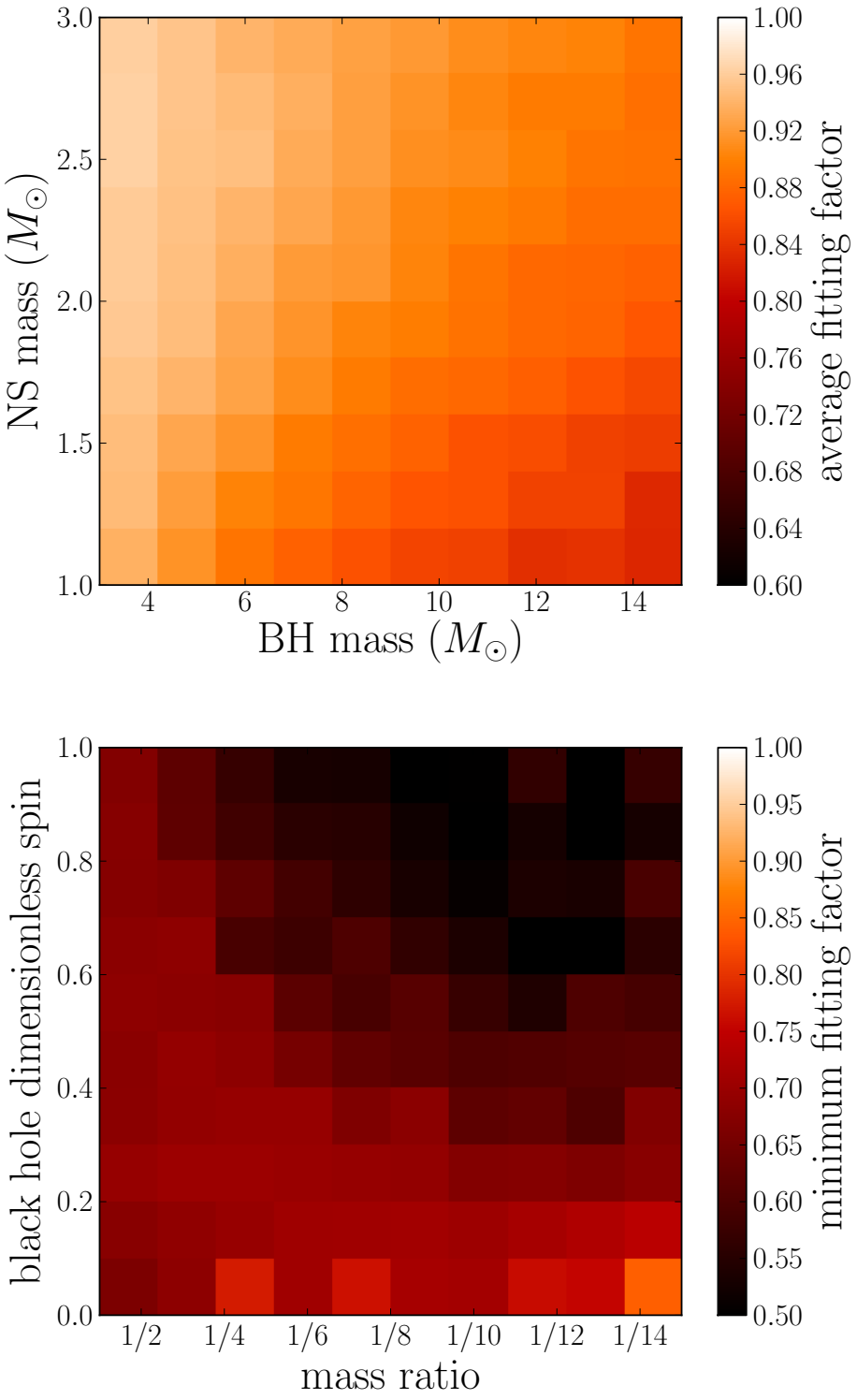

Figure 38: Average fitting factor between a set of generic, precessing, NSBH signals and a templa
masses (top left) and as a function of the mass ratio and the black hole dimensionless spin magnit
left) and the signal recovery fraction (bottom right) as a function of the mass ratio and the blac
waveforms are modelled using the TaylorT4 approximant. The distribution that the NSBH signa
construction is described in section 4.7. Results obtained using the zero-detuned, high-power adv

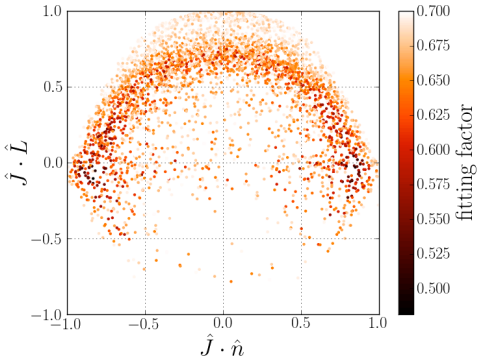

Figure 39: The distribution of precessing NSBH signals that are recovered with fitting factors < 0.7 when searching with an aligned-spin template bank. We use $\hat{J}$ to denote the initial total angular momentum of the system, $\hat{n}$ denotes the line of sight towards the observer and $\hat{L}$ denotes the orbital angular momentum when the gravitational wave frequency is 60 Hz (at which point approximately half of the signal power has accumulated). Both signals and template waveforms are modelled using the TaylorT4 approximant. The distribution that the NSBH signals are drawn from is described in section 4.3. The template bank construction is described in section 4.7. Results obtained using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

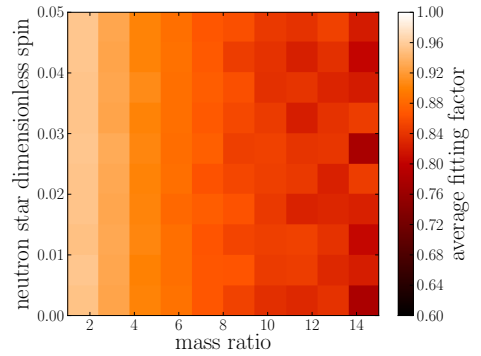

Figure 40: Average fitting factor between a set of generic, precessing, NSBH signals and a template bank of aligned-spin waveforms as a function of the mass ratio and the neutron star dimensionless spin magnitude (top right). Both signals and template waveforms are modelled using the TaylorT4 approximant. The distribution that the NSBH signals are drawn from is described in section 4.3. The template bank construction is described in section 4.7. Results obtained using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

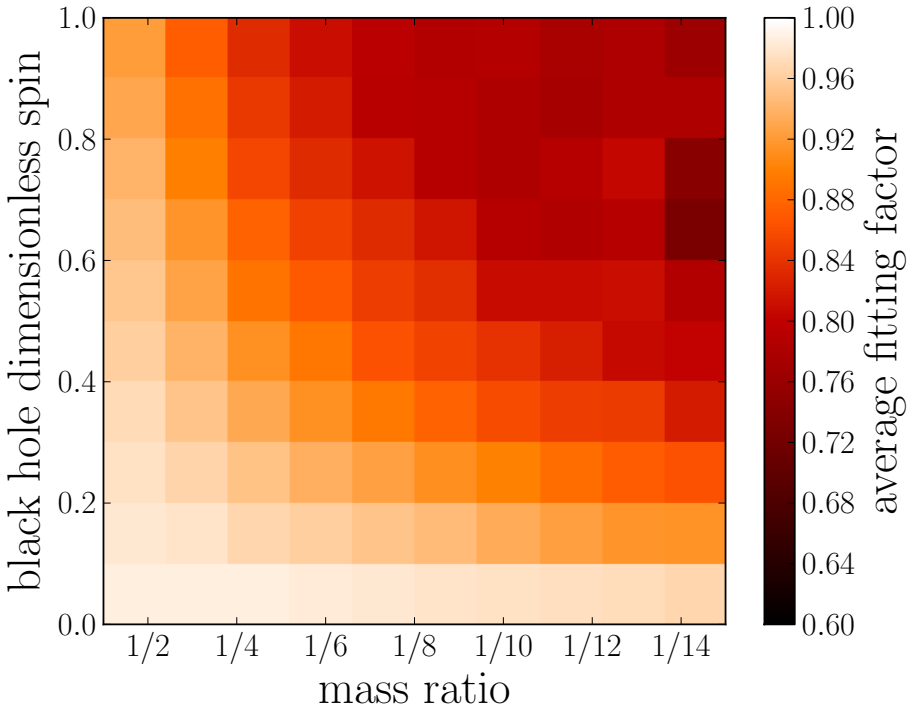

Figure 41: Average fitting factor between a set of generic, precessing, NSBH signals and a templat
the black hole dimensionless spin magnitude. Shown when both the template waveforms and s
waveforms are modelled with TaylorT2 and the signals are modelled with TaylorT4 (right). The re
right panel of Fig. 38. The distribution that the NSBH signals are drawn from is described in sect
Results obtained using the zero-detuned, high-power advanced LIGO sensitiv

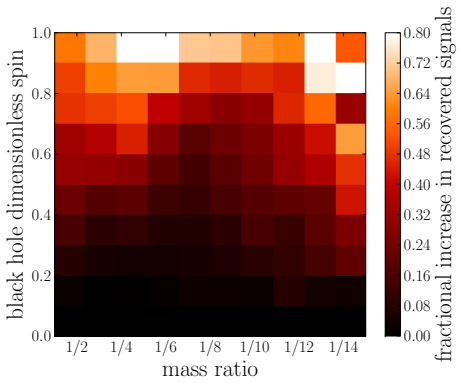

Figure 42: The fractional increase in the number of recovered signals when searching for generic, precessing, NSBH signals using a template bank of aligned-spin waveforms and a template bank of non-spinning waveforms. Both signals and template waveforms are modelled using the TaylorT4 approximant. The distribution that the NSBH signals are drawn from is described in section 4.3. The template bank construction is described in section 4.7. Results obtained using the zero-detuned, high-power advanced LIGO sensitivity curve with a 15Hz lower frequency cut off.

caused by the discreteness of the bank, are due only to the effects of precession. In both cases we observe a median fitting factor of ~ 0.95 and a mean fitting factor of ~ 0.91. This is a clear improvement over the non-spinning results where the mean fitting factor was 0.82 (0.84) for TaylorT2 (TaylorT4) and the median fitting factor was 0.86 (0.88).

In Fig. 34 we also show results where the template waveforms are modelled with TaylorT2 and the signals are modelled with TaylorT4. In this case the performance is worse, with a median fitting factor of ~ 0.92 and a mean fitting factor of ~ 0.88.

In Fig. 38 we show the mean fitting factor as a function of the intrinsic parameters for our results with the TaylorT4 waveform. We also show the minimum fitting factor and the signal recovery fraction as a function of the BH spin magnitude and mass ratio for the same results. The Figure serves to highlight that there are certain systems in certain regions of the parameter space where precessional effects cause the NSBH signals to have large mismatches with a bank of aligned-spin templates. This is most prominent when m_{BH}/m_{NS} and the BH spin magnitude are both large, ie. where the black hole's angular momentum is particularly large relative to the orbital angular momentum. We explore this further in Fig. 39 where, following the work of [**?**], we

show the distribution of precessing systems recovered with fitting-factors smaller than 0.7. This is plotted as a function of the angles between the total angular momentum, the orbital angular momentum and the line of sight to an observer. As predicted in [**?**], there is clearly a correlation between these angles and the systems recovered with the lowest fitting factors. To demonstrate that these results are not specific to the TaylorT4 waveform, in Fig. 41 we show the mean fitting factor as a function of the BH spin magnitude and mass ratio for our TaylorT2 vs TaylorT2 and TaylorT2 vs TaylorT4 results. The TaylorT2 results are very similar to the TaylorT4 results in Fig. 38. This again demonstrates that the choice of waveform is not affecting our statements regarding the effect precession will have on searches for NSBH signals using aligned-spin template banks. When searching for TaylorT4 signals with TaylorT2 templates we see lower fitting factors. The disagreement between these two waveform models is a significant factor that will affect searches for NSBH systems with second generation observatories. Computing higher order terms in the Post-Newtonian (PN) expansion of the center-of-mass energy and gravitational wave flux will help to reduce this disagreement and produce waveforms that better match real gravitational-wave signals. In Figure 40 we plot the average fitting factor as a function of the mass ratio and the *neutron star* dimensionless spin. There is not any noticeable correlation between the average fitting factor and the neutron star's spin.

We can also compare these results to the results we obtained using a non-spinning template bank in section 4.9.1. In Fig. 42 we show the fractional increase in the number of recovered signals between using non-spinning and aligned-spin template banks for the TaylorT4 approximant. The fractional increase in the number of recovered signals is calculated by taking the ratio of the signal recovery fraction when using a non-spinning bank and the signal recovery fraction when using an aligned-spin bank. This figure helps to emphasize that a much greater fraction of systems with large spin would be recovered when using an aligned-spin template bank. In Table 2 we summarize the average signal recovery fractions for the aligned-spin banks and compare these numbers to the results obtained with non-spinning template banks. We remind the reader that we are comparing signal recovery at a fixed signal-to-noise ratio. Signal recovery at a fixed false-alarm probability will depend on other factors, including the size of the parameter space covered by the template bank and the non-Gaussianity of the data. We discuss this further in the conclusion.

Template approximant	Signal approximant	Signal recovery fraction for non-spinning bank		Signal recovery tion for aligned bank	
		Average	$(10, 1.4)M_\odot$	Average	(10
TaylorT2	TaylorT2	64%	63%	83%	74%
TaylorT4	TaylorT4	69%	67%	82%	73%
TaylorT2	TaylorT4	67%	64%	77%	67%

Table 2: The performance of our aligned-spin template banks when used to search for a set of ge template and signal waveforms. We show both the mean signal recovery fraction over the full NSBH NSBH system with masses $(10 \pm 0.5, 1.4 \pm 0.05)M_\odot$. The distribution that the NSBH signals are dra is described in section 4.7. Results obtained using the zero-detuned, high-power advanced LIGO s upper frequency cut off.

Finally, we compare our results with previous works. In [**?**] the authors presented an efficiency study when using a template bank of stochastically generated aligned-spin signals. We verified that when using the stochastic algorithm we used in this work, and using the same set of parameters as the study described in [**?**], we generated a bank with the same number of templates. We have therefore demonstrated that our template bank algorithm requires less templates to acheive the same level of coverage as the algorithm used in [**?**]. In that work the effective fitting factor for a NSBH system with masses given by $10M_\odot$, $1.4M_\odot$ was estimated to be 0.95, which corresponds to a signal recovery fraction of 86%. In contrast, our results show a lower signal recovery fraction for the same masses of $73 - 74\%$ when the same waveform model is used to model both the template and signal. It isn't clear why this discrepancy occurs, however it may be partially explained by the fact that the authors of [**?**] used a lower frequency cutoff in their matched-filters of 20Hz, whereas we used 15Hz, which is more appropriate for the predicted aLIGO zero-detuned–high-power noise curve.

In [**?**] the authors used a simplified model of precessing systems to predict the distribution of fitting factors for NSBH systems. These results, shown in Figure 11 of that work, agree qualitatively with the results obtained here. We also obtain quantitative agreement by comparing our simulations of generic precessing systems with TaylorT4 as the signal and template model with the values predicted by Eq. 46b of [**?**]. We find that 90% of the fitting factors are within 0.03 of the predicted values. They also predicted the distribution of the signals that would be recovered with the lowest fitting factors as a function of the orientation of the black hole spin

and the orientation of the orbital plane with respect to the line of sight. We produce a similar distribution in Fig. 39. A further exploration of the agreement of the fitting factors with this prediction will be carried out in a future work making use of these simulations.

4.10 Conclusions

In this work we have explored the effect that the angular momentum of the black hole will have on searches for neutron-star black-hole binaries with aLIGO. The black hole's angular momentum will affect the phase evolution of the emitted gravitational-wave signal, and, if the angular momentum is misaligned with the orbital plane, will cause the system to precess. We have found that if these effects are neglected in the filter waveforms used to search for NSBH binaries it will result in a loss in detection rate of $31-36\%$ when searching for NSBH systems with masses uniformly distributed in the range $(3-15, 1-3)M_{\odot}$. When restricting the masses to $(9.5-10.5, 1.35-1.45)M_{\odot}$ we find that the loss in detection rate is $33-37\%$. The error in these measurements is due to uncertainty in the PN waveform models used to simulate NSBH gravitational-wave signals. In a companion work we investigate how the uncertainty in waveform models used to simulate NSBH waveforms will reduce detection efficiency [**?**].

We have presented a new method to create a template bank of NSBH filter waveforms, where the black hole's angular momentum is included, but is restricted to be (anti-)aligned with the orbit. These waveforms will include the effect that the black hole's angular momentum has on the phase evolution of the gravitational-wave signal, but will not include any precessional effects. We have shown that this bank offers a $16\%-30\%$ improvement in the detection rate of neutron-star black-hole mergers when compared to a non-spinning template bank when searching for NSBH systems with masses in the range $(3-15, 1-3)M_{\odot}$. However, when searching for NSBH systems with masses restricted to the range $(9.5-10.5, 1.35-1.45)M_{\odot}$ we find the improvement is reduced to $5-17\%$. Some systems are not recovered well with this new bank of filters. These systems are ones where the black-hole spin is misaligned with the orbit and the waveform is significantly modified due to precession of the orbital plane. This happens most often when m_{BH}/m_{NS} and the spin magnitude are

both large. In [**?**] the authors predict where in the parameter space to expect NSBH systems that will not be recovered well by non-precessing template banks. These predictions were given in terms of the angles between the orbital plane, the black hole's angular momentum and the line-of-sight to an observer. These predictions agree with the results that we obtain in this work. In [**?**] the authors claim that an aligned-spin template bank will be effectual for detecting precessing NSBH systems. In this work, we find that with an aligned-spin template bank $17 - 23\%$ of NSBH systems will be missed compared to an ideal search with exactly matching filter waveforms. In reality this ideal search could never be performed as it would require an infinite number of filter waveforms. Template banks are usually constructed to allow for no more than a 3% loss in SNR, therefore we expect to lose up to 10% of systems even if the template bank fully covers the signal parameter space. We therefore conclude that searches using precessing waveforms as templates could potentially increase the detection rate of NSBH signals, but not by more than $\sim 20\%$. Performing such a search would, however, remove an observational bias against systems where precessional effects are most prevalent in the gravitational-wave signal.

These figures are also affected by the parameter distribution chosen for the NSBH systems. Here we chose a distribution that is uniform in mass, uniform in spin magnitudes, isotropic in spin orientations and isotropic in orientation parameters and sky location. We have however, explored how the ability to detect precessig NSBH signals varies as a function of the masses and spins as seen in Figures 38 and 39.

When searching for NSBH systems in aLIGO one has to consider the non-Gaussianity of the background noise, which we have not done in this work. A non-Gaussian noise artifact can produce SNRs that are considerably larger than those expected from Gaussian noise fluctutations. To deal with this, numerous consistency tests are used in the analyses to separate gravitational wave signals from instrumental noise artifacts [**?**]. It is possible that the detection rate could be further reduced from the values we quote in this work if some signals *fail* these consistency tests and are misclassified as non-Gaussian noise transients. However, these signal consistency tests should only act to remove, or reduce the significance of, events that already have low fitting factors and therefore do not match well with the search templates. Another important consideration is that of the number of templates used in the bank. To

achieve higher fitting factors will require more template waveforms, covering a larger signal space, which will allow more freedom in matching the background noise and will mean that the SNR of the loudest background triggers will increase. Therefore signals will need slightly higher SNRs to achieve the same false alarm probability. However, a factor of 10 increase in the number of *independent* templates will only increase the expected SNR of the loudest background event by less than 5%, if Gaussian noise is assumed. Therefore, while we are careful to note these considerations, we do not believe they will have a large impact on the numbers we quote above and leave a detailed investigation of such effects to future work.

In this work we have restricted ourselves to considering post-Newtonian, inspiral-only signal waveforms and consider only the case of two point particles. This was done as there is not currently any widely available waveform model that includes both the full evolution of a NSBH coalescence *and* includes precessional effects over the full parameter space that we consider. When such a model is available it may be that tidal forces and the merger component of the waveform may affect our conclusions. We believe that such effects will be limited as the black hole mass is $< 15M_{\odot}$ in our simulations, however it would be informative to repeat our simulations when a full NSBH waveform model is available.

Acknowledgements

We thank Stefan Ballmer, Alessandra Buonanno, Eliu Huerta, Prayush Kumar, Richard O'Shaughnessy, B. S. Sathyaprakash, Peter Saulson, Matt West and Karl Wette for useful discussions. We also thank Frank Ohme and the anonymous referee for providing thoughtful and insightful comments on this manuscript. This work is supported by National Science Foundation awards PHY-0847611 (DAB, AHN), PHY-1205835 (AHN, IWH), PHY-0970074 (EO), PHY-0855589 (AL) and PHY11-25915 (DAB,IWH,EO,AL). DAB, IWH, AL, and EO thank the Kavli Institute for Theoretical Physics at Santa Barbara University, supported in part by NSF grant PHY-0551164, for hospitality during this work. DAB thanks the LIGO Laboratory Visitors Program, supported by NSF cooperative agreement PHY-0757058, for hospitality. DK and AL thank the Max Planck Gesellschaft for support. DAB is supported by a Cottrell Scholar award from the Research Corporation for Science Advancement.

Computations used in this work were performed on the Syracuse University Gravitation and Relativity cluster, which is supported by NSF awards PHY-1040231 and PHY-1104371.

Chapter 5

PyCBC Optimization

5.1 Introduction

Compact binary coalescence is the most promising source of gravitational-waves for Advanced LIGO [**?**]. The inspiral and merger of a binary containing stellar-mass compact objects (neutron stars and black holes) generates gravitational waves that sweep upward in frequency and amplitude through the sensitive band of Advanced LIGO. Compact binary coalescence searches in Initial LIGO [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] were performed by a high-latency (or deep, offline) search pipeline. For Advanced LIGO, both high-latency offline deep searches and low-latency rapid-result searches will be performed. The offline search pipeline performs: (i) initial analysis of LIGO data with and without simulated signals to measure detector performance and tune the parameters for a deep search; (ii) full analysis of the data set with the final search parameters to detect signals and measure the false alarm rate of detection candidates; and (iii) reanalyzing the data with the addition of a large number of simulated signals to determine LIGO's sensitivity and selection bias to the astrophysical population. The high-latency CBC search incorporates information not available in the low-latency search (for example, improved detector calibration information and data quality information from offline detector characterization). It also calculates detection probabilities using large, time-symmetric samples of the detector noise background and performs a deep, comprehensive search over the full parameter space of compact binaries. The offline search measures the selection bias of the CBC search, including the loss in detection efficiency caused by uncertainties in the model waveforms, or

Astrophysical search target	**E5-2670 MSU per year**		
	2015–16	**2016–17**	**2017–18**
Highest priority: Binary neutron stars (non-spinning templates)	0.084	0.514	1.48
High priority: Binary neutron stars (aligned-spin templates)	1.25	8.82	30.2
May 2014 Request: Binary neutron stars (aligned-spin templates)	*44.4*	*130*	*270*
Highest priority: Neutron star–black hole (aligned spin templates)	2.26	17.5	65.7
May 2014 Request: Neutron star–black hole (aligned spin templates)	*47.1*	*167*	*494*
Highest priority: Binary black hole search (aligned spin templates)	1.45	11.9	36.1
May 2014 Request: Binary black hole search (aligned spin templates)	*23.5*	*72.3*	*151*
Total for all high-latency CBC searches	**5.04**	**38.7**	**133**
May 2014 Total for all high-latency CBC searches	*115*	*369*	*915*

Table 3: The computational resources needed to achieve the LVC's production high-latency CBC search in millions of service units (MSU) per year. One service unit is defined as one core hour on an Intel® E5-2670. Shown below each science goal (in italics) is the size of the corresponding May 2014 request from Table 3 of LIGO-T1400269; the difference from May 2014 to April 2015 reflect optimization of code and the flow down of science priorities. Since no search pipeline for precessing searches exists, this search is currently not listed in the LSC's prioritized science goals. Large-scale simulations included in the above cost will allow us to measure the sensitivity of aligned-spin searches to precessing systems.

the omission of certain physics (e.g. binary precession) in the detection templates. In the absence of a detection in a given region of the parameter space, the deep offline search pipeline computes upper limits on the rate of CBC sources which can be used to constrain astrophysical models of binary and compact object formation. Consequently, both the deep, offline and the low-latency pipelines are required to achieve Advanced LIGO's science goals.

This document describes the computational cost of the high-latency CBC search, which has been substantially re-written for Advanced LIGO. The prioritized request for computing for binary neutron star, neutron star–black hole, and binary black hole sources is summarized in Table 3 (in Intel® E5-2670 core hours per year). For reference, we include the request from Table 3 of LIGO-T1400269 presented in May 2014[1].

We note that the computational resources requested here for the high-latency CBC

[1]The May 2014 offline CBC request in LIGO-T1400269 was presented in Stampede SU, which assumes that a core is an Intel® E5-2680. For direct comparison in Table 3, we convert this request to E5-2670 cores by multiplying Stampede SU by the ratio of the clock speeds, i.e. 2.7/2.6.

request are significantly less than those presented in the May 2014 request, with the 2015/16 request being a factor of 23 smaller and the 2017/18 request being a factor of 6.8 smaller. The majority of this reduction is due to the significant improvements we have made in the new PyCBC search executable, compared to the old LALApps search code. **The new PyCBC code is a factor of 6.75 faster in terms of search throughput on our reference CPU platform than search code used in Initial LIGO.** We describe the optimizations that we have made to achieve this in detail in Section 5.5.

Further reductions are due to changes in the size of the template bank used in the search, which changes computational cost linearly. As a result of the scientific prioritization process and input from the astrophysics community, we increased the minimum neutron star mass in the binary neutron star and neutron star–black hole searches from $0.9\,M_\odot$ to $1.0\,M_\odot$. This results in a template bank that is a factor of ~ 1.3 smaller than that used in May 2014. Further reductions in the size of the template bank are due to the use of a more refined noise curve than that used in May 2014 to model the detector sensitivity. Template banks computed using the more accurate noise model are a factor of 1.1–2 times smaller. We have also used a more realistic estimate of the detector observation time, which reduces the requested computational resources by a factor of 2.4 in 2015–16, 1.8 in 2016–17, and 1.4 in 2017–18. We have also determined that a larger number of simulated signals will be required to measure the efficiency of the offline search than accounted for in the May 2014 request. This increases our request by a factor of 1.6 in 2015–16, 2 in 2016–17, and 2.5 in 2017–18. However, the overall cost is still substantially smaller than that presented last year.

There are several uncertainties in the computational cost estimates presented here. The most significant uncertainty is the detector sensitivity and bandwidth, as described in Section 5.6. If detector commissioning progresses at a more rapid pace and we achieve the best expected Advanced LIGO sensitivity in O3, our computational cost would increase by $\sim 50\%$. The throughput of the search code also depends on the (as yet unknown) stationarity of the detector data, with more stationary data having a faster throughput. If the data are very clean, then the computational cost could be $\sim 20\%$ less than requested here. If the data are very non-stationary, containing many non-Gaussian noise transients, then the computational cost could increase by $\sim 40\%$.

We will continue to refine our computational needs based on instrument progress and our best known predictions for the detector's sensitivity evolution.

In addition to our optimization work on CPUs, we have also explored the use of GPUs, which show significant promise for use in the high-latency CBC search. The fastest throughput we have observed on a GPU with our initial CUDA kernels is a factor of ~ 3 higher than the fastest throughput on a CPU socket. We have also explored the use of consumer grade GPUs and have demonstrated that these can yield an order of magnitude greater throughput per dollar than CPUs. Our initial GPU kernels have not yet been fully optimized. We are collaborating with NVIDIA to increase their performance, as described in Section. 5.7.1.

Finally, we note that the scientific methods used for computational cost estimates here are the same as those used for searches for gravitational waves in Initial LIGO. Reduction in computational costs described below therefore result from optimization of the existing methods, prioritizing our science, and better estimates of detector performance, rather than replacement of Initial LIGO scientific methods with new ones. We have demonstrated that the optimized code reproduces the results obtaining in Initial LIGO, but at substantially reduced computational cost. This gives us confidence that the optimized code described here will be successful in Advanced LIGO as we explore new methods that may further reduce the cost and allow exploration of larger parameter spaces (in particular, the space of precessing binaries).

The rest of this document is organized as follows: Section 5.2 reviews the high-latency CBC search pipeline and Section 5.3 describes the scientific methods that we have investigated to implement the high-latency CBC search. Section 5.4 describes the data-analysis methods that dominate the computational cost of the search pipeline (the matched filter and the time-frequency signal-based veto). Section 5.5 discusses the selection of optimal algorithms, libraries, and tests of our implementation on CPU hardware. In particular, Section 5.5.1 describes an improved implementation of the time-frequency signal consistency test, and Section 5.5.1 describes an improved algorithm for event finding and clustering in the matched filter output. Both of these improvements are independent of hardware implementation. We then focus on optimization on the LIGO reference CPU, the Intel® E5-2670 (which is similar to the E5-2680 used in Stampede). Section 5.5.2 describes the selection of optimal FFT engines for this hardware, and Sections 5.5.2 and 5.5.2 describe the improvements to

parallelize and vectorize the non-FFT portions of the filtering code. With these improvements, we find that the fastest FFT method (eight core multi-threaded FFTW) provides the greatest search throughput. Section 5.5.2 compares the best measured performance with our theoretical expectations. Based on the fastest CPU implementation on the E5-2670, and taking into account instrument and astrophysics tuning, Section 5.6 calculates the resources required for the production high-latency CBC searches, as summarized in Table 3. Finally, Section 5.7 describes our hardware trade study investigating performance on available CPU systems and, in Section 5.7.1, our implementation of the high-latency search on Graphics Processing Units.

5.2 Compact Binary Coalescence Searches

If the angular momenta of the compact objects—their *spins*—are aligned with the orbital angular momentum of the binary (or the compact objects are non-spinning), then the gravitational-wave strain h observed by LIGO or Virgo (neglecting higher-order amplitude corrections) can be written as

$$h(t-t_c) = A(t-t_c)\cos(\phi(t-t_c))\cos\Phi - A(t-t_c)\sin(\phi(t-t_c))\sin\Phi, \qquad (5.1)$$

where t_c is the coalescence time of the binary, $A \propto f_{\mathrm{GW}}^{2/3}$ is the amplitude of the wave, Φ is a constant that depends on the orientation of the binary, and ϕ is the time-evolving gravitational-wave phase—the quantity to which LIGO and Virgo are most sensitive. The gravitational-wave phase evolution is given by the particular waveform model used in the search and depend only on the masses and spins of the compact objects. For detection of binaries with total mass $M \lesssim 12M_\odot$ and spins $\chi \lesssim 0.4$ (which includes binary neutron stars), post-Newtonian theory provides a sufficiently accurate analytic model of the gravitational waveforms [**?**, **?**, **?**, **?**, **?**, **?**]. As the mass ratio and spins of the compact objects increase [**?**], or the total mass of the binary increases [**?**, **?**], post-Newtonian waveforms become less accurate. In this case, we can model the signal waveforms using analytic models methods, such as the effective one body (EOB) approach [**?**] tuned to numerical relativity simulations of binary black holes [**?**], or by phenomenological models that capture the dynamics of binary black holes [**?**, **?**].

The amplitude of gravitational waves measured by the Advanced LIGO and Virgo

detectors is expected to be comparable to (or smaller than) the mean amplitude of the detector noise. Consequently, digital signal processing is required to extract signals from the noisy detector data. For CBC sources, it is possible to construct models of the gravitational waveform. Matched filtering [**?**] would be the optimal method to identify signals in the detector data, if the detector noise were stationary and Gaussian. The noise in the LIGO and Virgo detectors from fundamental sources (thermal noise, radiation pressure noise, and photon shot noise) does behave in this way; however populations of non-Gaussian, non-stationary transients of both environmental and instrumental origin are also present. These "glitches" cause excursions in the matched filter SNR that may be mistaken for signals. Requiring that a signal be seen in both LIGO detectors (and once running the Virgo detector) eliminates a substantial fraction of false signals; however additional waveform consistency tests [**?**] are needed to determine if SNR excursions are due to a glitch or a gravitational wave [**?**].

The gravitational waveform depends sensitively on the mass and spin parameters of the source. The parameters of a signal are not known in advance, so a discrete "bank" of gravitational-waveform templates [**?**, **?**, **?**, **?**, **?**] is constructed that is sensitive to the target astrophysical population. The computational cost of the search scales effectively linearly with the number of templates in the bank, as the matched filter is applied once per template in the bank. Template banks exist for binaries where the angular momentum of the compact objects is negligible (non-spinning binaries) [**?**, **?**] and for the case in which the component object's spin is aligned with the orbital angular momentum of the binary [**?**, **?**]. If the spin of the compact object is not aligned with the orbital angular momentum (for example due to misalignment due to a supernova kick), spin-orbit coupling will cause the plane of the binary to precess. Spin-orbit and spin-spin coupling also change the rate of energy and angular momentum loss for the binary, which further changes the gravitational-wave signal. For binary neutron stars, a search for aligned-spin systems is sufficient to capture precessing systems, as the effects of precession are not significant for these systems [**?**]. However, for binary black holes or neutron star–black hole binaries spin and precession effects can be significant. At present no template placement algorithm or search pipeline has been implemented for spinning, precessing BBH or NSBH binaries. Search methods which incorporated spin effects were considered in Initial LIGO, but were found to increase the false alarm rate resulting in a less sensitive

search [?, ?, ?]. Development of precessing binary searches is an active topic of research, but these pipelines are not yet in production for the LIGO-Virgo searches. In the absence of a search for precessing binaries, simulated signals from a population of precessing binaries will allow us to quantify the sensitivity of the LIGO-Virgo current search to the astrophysical population.

The full search for coalescing compact binaries requires: (i) generation of the gravitational-wave template bank [?, ?]; (ii) filtering the data against this bank and identifying "triggers", or times where the matched filter SNR exceeds a certain threshold for a particular template [?]; (iii) checking triggers for consistency using waveform consistency tests [?]; (iv) folding in information from instrumental health and status information to further eliminate triggers due to instrumental artifacts [?, ?]; (v) applying coincidence algorithms to ensure that a gravitational-wave signal is present in two or more detectors with consistent signal parameters; and (vi) measuring the significance of a candidate signals by comparing their amplitude to that of the noise-induced background. Executing all of the above steps is the job of the *analysis pipeline* [?, ?], a program that generates a workflow that turns the raw detector data into a detection statement or measures the signal parameters. The search pipeline is a heterogeneous mixture of computational components that perform the required steps on all of the data in the correct order. LIGO-specific scripts write workflows that are planned by the Pegasus Workflow Management System into directed acyclic graphs (DAGs) that are executed by HTCondor's DAGman. The computational cost of the full pipeline is dominated by the executable that computes the matched filter and the waveform consistency tests. Based on a test filtering 11.5 days of simulated Advanced LIGO data, we find that 99.8% of the total 3727.63 core-days of runtime is spent in the `pycbc_inspiral` filtering engine (that computes the templates, the correlation, the FFT, and event finding and clustering), with the remainder spent in the coincidence, background estimation, and post-processing steps. Consequently, we have focused our optimization efforts to date on the filtering executable (although improvements have also been made to other parts of the pipeline).

A complication encountered when benchmarking the offline CBC search is that the run time of the search depends on the quality of the (random) detector noise. To save computational cost, the filtering executable only computes waveform consistency checks when the SNR exceeds a threshold value. However, we do not know in advance

the character or the rate of non-stationary noise transients in the data. Indeed, this rate can change over time during an observing run. To compute the computational cost of the analysis, we take the average of the measured throughput over three representative types of data.

5.3 Exploring the space of appropriate scientific methods

The scientific methods used in the pipeline may be implemented by more than one computational algorithm. For example, the matched filter signal-to noise ratio for a bank of templates may be constructed by a frequency-domain correlation or linear combinations of time domain correlations with an orthonormal filter set found via the singular value decomposition (SVD). Benchmarking of the LLOID algorithm [**?**] (which uses SVD and muti-rate filtering to implement a low-latency search as described in LIGO-T1400542) versus the frequency-domain FFT [**?**] has shown that the FFT method has a higher template-per-core throughput for a given input data sample rate. We have therefore selected the frequency-domain FFT correlation as the optimal scientific method to search for signals with the deep, offline pipeline where latency is not a concern. Since the LLOID method is used to search for signals with a possible rapid electromagnetic counterpart, it is appropriate to trade computational cost for latency in the low-latency search.

Non-Gaussian noise transients in the detector data can cause the matched filter to generate false triggers, and so a variety of *signal-based vetoes* have been developed that use additional information to distinguish signals from noise. These are often called χ^2-vetoes, as the three primary tests considered (known as the time-frequency signal-based veto, the autocorrelation signal-based veto, and the template bank veto) all generate a statistic that is χ^2 distributed in Gaussian noise. The low-latency CBC search implements the autocorrelation signal-based veto, as it is straightforward to compute in the low-latency search given its locality in time and dependence only on the SNR time-series. The offline search used in Initial LIGO CBC searches (and used for cost estimates in the May 2014 review) computed the time-frequency χ^2 veto and the bank veto. The time-frequency χ^2 veto has been demonstrated to be a powerful test and is essential to eliminate non-Gaussian transients from the search. However, it has not been demonstrated that the bank veto provides additional noise rejection

power beyond the time-frequency χ^2 veto. It has therefore been decided to eliminate the computation of the bank veto from the filtering to save computational cost. We are also exploring the use of the autocorrelation χ^2 veto (used in the low-latency search) in the deep, offline search. Preliminary investigations with prototype code suggest that this may provide additional information to the time-frequency χ^2 test for certain types of noise events, however it is not yet clear if this veto provides more noise-rejection power than the standard time-frequency χ^2.

Two further refinements of the scientific methods and tuning used in Initial LIGO are being explored: (i) In Initial LIGO, the search algorithm processed fifteen data segments of length 256 seconds though each template, with noise power spectral density (PSD) estimation performed over 2048 seconds. The new `pycbc_inspiral` executable allows us more control over the PSD estimation, as well as the number and length of the filter data segments. Preliminary investigations with increasing the length of the data segment to 512 seconds have shown a performance increase of $\sim 25\%$, as longer data segments allow us to make more efficient use of the matched filter output by decreasing the amount of time per data segment corrupted by the wrap-around of the FFT; (ii) Re-using the template for more data segments also further reduces the overall cost of the code. If Advanced LIGO provides sufficient long, stable lock stretches, we can increase the number of data segments per template, further reducing the overall search cost.

Moving beyond the existing methods, we are developing a new matched-filtering algorithm that performs a search that is hierarchical in sample rate. An approximate signal-to-noise time series is first created by reducing the sample rate, allowing the use of shorter, faster FFTs. Peaks of interest are found using a threshold that has been lowered to account for both the loss of the high frequency contribution to the SNR and the time offset from the full sample rate peak. Full sample rate matched filtering is performed at these peaks and the nearby points, to minimize the probability that the full sample rate peak is missed. This step is accelerated using a pruned FFT algorithm, where we decompose an N points FFT into a batched set of FFTs each $N^{(1/2)}$ in length, followed by an explicit DFT calculated for every point. A further optimization removes the first memory transposition of the FFT by storing the input data in a memory layout that is already transposed for the first batched FFT. This algorithm is efficient due to the small ratio between the number of interesting points

to calculate and the number of points in the full time series, $O(10^{-4})$, which allows us to make more efficient use of the Level 3 cache.

This method performs an accurate approximation to the full matched filter; however, it does not exactly reproduce the same output. Consequently, this method require careful testing before it can be commissioned as a production search. A prototype implementation has been tested in the binary black hole search, with the initial version of the code showing a 2–3 times speedup over the full sample-rate computation of the matched filter. Ongoing work to develop this method includes increasing the efficiency of the full sample rate reconstruction which will allow this algorithm to be efficient in the widest parameter space range. If this new method can be demonstrated to yield the same detection efficiency as the current methods, we will adopt it for Advanced LIGO, further reducing computational cost.

5.4 Computational Methods

The computational cost of the CBC search is dominated by the FINDCHIRP matched filtering algorithm [**?**] that computes the matched-filter SNR and the waveform consistency test for a single detector; combining these triggers from multiple detectors is relatively inexpensive. The matched-filter SNR ρ^2 for the data s and template h, analytically maximized over A and Φ, is given by

$$\rho^2 = \frac{(s|h_0)^2 + (s|h_{\pi/2})^2}{(h_0|h_0)}; \quad \text{with} \quad (a|b) = 4\,\mathrm{Re}\int_{f_{\mathrm{low}}}^{f_{\mathrm{high}}} \frac{\tilde{a}(f)\,\tilde{b}^*(f)}{S_n(f)} df, \tag{5.2}$$

where $S_n(f)$ the one-sided detector-noise PSD and h_0 and $h_{\pi/2}$ correspond to the two gravitational-wave polarizations. If a gravitational wave signal is present in the data, then its location in time is defined by the parameter t_c. To search over all possible times t_c, we use a Fast Fourier Transform to compute the value of the inner product $(s|h_0)$ by

$$(s|h_0(t_c)) = 2\int_{-\infty}^{\infty} df e^{2\pi i f t_c} \frac{\tilde{s}(f)\tilde{h}_0^*(f)}{S_n(|f|)} \tag{5.3}$$

and the square of the SNR for a chirp that ends at time t is

$$\rho^2(t) = \frac{1}{(h_0|h_0)}\left[(s|h_0(t))^2 + (s|h_{\pi/2}(t))^2\right] \equiv \frac{1}{\sigma^2}\left[\rho_0^2(t) + \rho_{\pi/2}^2(t)^2\right] \tag{5.4}$$

where the two SNR time series $\rho_0^2(t)$ and $\rho_{\pi/2}^2(t)$ can be obtained by inverse Fourier transforms of the form in Eq. (5.3). The FINDCHIRP algorithm incorporates several optimizations to compute the matched-filter SNR. FINDCHIRP assumes that the two chirp waveforms $\tilde{h_0}$ and $\tilde{h}_{\pi/2}$ are orthogonal. This is identically true for the aligned-spin waveforms used in BNS and NSBH searches, and approximately true for the slowly-evolving inspiral part of BBH waveforms. The filtering cost is reduced by packing the two filter phases into the real and imaginary components of a single complex inverse FFT rather than computing it independently from two real inverse FFTs. For BNS and NSBH waveforms, we use the stationary phase approximation to write the waveform directly in the frequency domain [**?**], eliminating the cost of Fourier transforming the waveform. FINDCHIRP further increases efficiency when using frequency-domain templates by splitting the filter into a part that depends on the data and a part that depends only on the template parameters, and reuses a template for several (typically 15) data segments before generating the next template. These optimizations further reduce the cost of computing the integrand of the matched filter [**?**]. The computational cost of the matched filter is therefore dominated by the complex vector multiplication needed to compute the integrand of Eq. (5.3) and the complex inverse FFT used to compute the SNR as a function of signal arrival time. A further step to find and cluster peaks in the SNR time series is computationally cheap, but may may be a performance bottleneck if not implemented carefully, particularly if multi-core FFTs are used to compute the matched filter.

Non-Gaussian noise transients in the detector may cause high SNR excursions which the matched filter alone cannot distinguish from signals. To distinguish a high SNR due to a signal from one due to a glitch, we use a time-frequency signal-based veto known as the time-frequency signal-based χ^2 veto [**?**]. This test divides the two template phases h_0 and $h_{\pi/2}$ into p frequency sub-intervals $\{h_0^l\}$ and $\{h_{\pi/2}^l\}$, $l = 1 \ldots p$ with

$$(h_0^l|h_0^m) = \frac{1}{p}\delta_{lm},\ (h_{\pi/2}^l|h_{\pi/2}^m) = \frac{1}{p}\delta_{lm},\ (h_0^l|h_{\pi/2}^m) = 0 \tag{5.5}$$

and $h_0 = \sum_{l=1}^{p} h_0^l$ and $h_{\pi/2} = \sum_{l=1}^{p} h_{\pi/2}^l$. We can then construct the $2p$ quantities $\{\rho_0^l\} = (s|h_0^l)$ and $\{\rho_{\pi/2}^l\} = (s|h_{\pi/2}^l)$, where where s is the detector output. The χ^2

test is constructed by computing

$$\chi^2 = p\sum_{l=1}^{p}\left[(\Delta x_l)^2 + (\Delta y_l)^2\right] \quad \text{where} \quad \Delta x_l = \rho_0^l - \frac{\rho_0}{p} \text{ and } \Delta y_l = \rho_{\pi/2}^l - \frac{\rho_{\pi/2}}{p}. \quad (5.6)$$

In the presence of Gaussian noise $s = n$ this statistic is χ^2 distributed with $\nu = 2p-2$ degrees of freedom. Furthermore, if a signal is present along with Gaussian noise $s = h + n$, then $\chi^2 = pr^2$ is still χ^2 distributed with $\nu = 2p - 2$ degrees of freedom. Small values of the χ^2 veto mean that the SNR has been accumulated in a manner consistent with an inspiral signal. Since the value of the χ^2-veto is only computed when peaks in the SNR are detected, the total computational cost depends on the noise content of the input data, which is non-deterministic.

5.5 Identifying computational algorithms that efficiently implement the scientific methods

In this section, we consider the optimizations that we have made to to the scientific methods selected. In Initial LIGO, searches were performed with the *ihope* search pipeline. The ihope pipeline was developed over the six initial LIGO science runs based on our experience searching kilometer-scale interferometric detector data for the first time. To provide a computational cost estimate for the May 2014 NSF review, we analyzed two weeks of Initial LIGO/Virgo data with the ihope search pipeline, in the configuration that was most recently used in the the S6/VSR2,3 science run [**?**, **?**]. The dominant computational cost of the ihope pipeline is the `lalapps_inspiral` filtering engine. Over the last two years, the ihope pipeline has been re-written for Advanced LIGO. The new framework, known as PyCBC is more modular, flexible, and scalable than the LALApps framework used previously. PyCBC has been developed to accommodate longer templates and larger template banks necessitated by the improved detector noise profile, as well as the lessons learned from the May 2014 NSF review and our optimization experience over the last year.

The PyCBC architecture implements the high-level program control in Python, however computations are performed using C code compiled just-in-time by the `scipy.weave` framework [**?**]. This ensures that all computationally intensive parts of the pipeline are executed by low-level, optimized code and not by the Python

interpreter. Furthermore, direct AVX/SSE calls or OpenMP parallelization may be performed by use of the X86 intrinsic functions in the weave-compiled C-code. The Python frame work allow us to modularize the low-level kernels at low overhead. It is therefore straightforward to replace these kernels with code for new compute architectures including Graphics Processing Units (GPUs) and Intel® MICs (in addition to architecture-specific CPU code) in the same search engine. This modularization reduces the human cost of development, validation, and verification, which is a concern given the small size of the development team (approximately 4 FTEs).

As a result of this development, the the `lalapps_inspiral` filtering engine has been retired and replaced with the new `pycbc_inspiral` executable. Our computational costs for Advanced LIGO are computed with the best current version of `pycbc_inspiral` on CPUs and GPUs; however, we provide some benchmarking for `lalapps_inspiral` in Appendix 5.8 to illustrate changes between the current code and the numbers presented in May 2014. In the sections below, we review the optimizations that have been made to the CPU-based code to obtain our current performance numbers.

5.5.1 Algorithmic Optimizations

In this section, we discuss improved algorithms that implement the selected scientific methods. By refactoring the code used to implement the time-frequency χ^2 signal-consistency test and the event finding and clustering, we have made performance improvements that can be realized independent of the architecture used (CPU or GPU). The improved algorithms generate *exactly the same output* as used in previous LIGO searches, as compared to a different choice of scientific method which may implement a somewhat different search (e.g. the hierarchical methods discussed above).

Optimization of the χ^2 signal-consistency test

If points are found above threshold in the matched filter signal-to-noise time series, then the a time-frequency signal consistency test is applied. The test consists of breaking the waveform into p frequency bins of equal power. Each bin is filtered against the data to obtain the partial SNR contribution ρ_l and then compared to the

expected SNR contribution ρ/p. This is expressed as

$$\chi^2 = p \sum_{l=0}^{p} [\rho_l - \rho/p]^2, \tag{5.7}$$

The calculation of each bin p requires a single FFT, and neglecting lower order terms, we find a cost of

$$\mathrm{FLOP} = p \times 5N \log(N). \tag{5.8}$$

The `lalapps_inspiral` implementation of this test computed the χ^2 time series for the *entire* data segment, if any points in the matched filter time series exceeded the threshold. This is computationally efficient, if there are many threshold crossings. However, if only a few points cross the threshold, then computation is wasted computing the χ^2 veto at unnecessary times. As we know the location of peaks in the SNR time series, we can directly calculate the χ^2 test only for those points. We can express the quantity that needs to calculated in terms of existing information as,

$$\frac{\chi^2 + \rho^2}{p}[j] = \sum_{l=0}^{p} \rho_l^2 \tag{5.9}$$

We can write this in terms of the quantities computed by the FINDCHIRP matched filter as

$$\frac{\chi^2 + \rho^2}{p}[j] = \sum_{l=0}^{p} \left(\sum_{k=k_l^{\min}}^{k_l^{\max}} \tilde{q}_k e^{-2\pi i j k/N} \right)^2 \tag{5.10}$$

where $\tilde{q}_k$ is the kernel of the matched filter (the frequency domain correlation of the noise-weighted template with the data), and the index k runs over frequency bins (typically $k_{\max} - k_{\min} \sim 10^5$). The quantity $[j]$ is the set of indices of the N_p peak values of the matched filter SNR. We note that the fact that $\tilde{q}_k$ is required in Eq. 5.10 is the reason that we use out-of-place FFTs for the matched filter. Computing Eq. 5.10 involves the calculation explicitly of $k_{\max} \sim 10^5$ twiddle factors[2]. This can be reduced to a complex multiply by calculating a single twiddle factor and iteratively finding

[2]The trigonometric constant coefficients that multiply the data.

the next factor, i.e.

$$\frac{\chi^2+\rho^2}{p}[j] = \sum_{l=0}^{p}\left(\sum_{k=k_l^{min}}^{k_l^{max}} \tilde{q}_k(e^{-2\pi ij/N})(e^{-2\pi ijk/N})^{k-1}\right)^2 \tag{5.11}$$

This reduces the computation cost to two complex multiples, one for the twiddle factor calculation, and one for the multiplication by $\tilde{q}$, along with a add of two complex numbers giving,

$$\text{FLOP} = 14 \times k_{\max} \times N_p \tag{5.12}$$

For small values of N_p we note that this can be vastly more efficient than the full FFT based calculation of the veto. The crossover point can be estimated as,

$$N_p = \frac{p \times 5N\log(N)}{14N_p k_{\max}}. \tag{5.13}$$

This algorithm has been implemented in the `pycbc_inspiral` filtering engine. The number of SNR threshold crossings is computed and the full χ^2 time series is calculated only if this number exceeds the above crossover point. Otherwise the χ^2 veto is computed in a point-wise manner at reduced cost. The exact cost reduction depends on the quality of the data, but on average, application of the approach on Initial LIGO data gives a reduction in computational cost of approximately a factor of four.

Optimization of thresholding and time clustering

After the matched filter SNR is computed for a given template, the resulting time series must be searched for points above a runtime-specified threshold to obtain gravitational-wave candidate triggers. Since both signals and glitches can produce many nearby SNR samples above threshold (which do not represent independent triggers), the SNR samples above threshold tend to be clustered in time. This leads to a high probability that there is a minimum spacing of a user-specified length (the clustering window) between any two consecutive clustered triggers. This window is chosen based on the impulse response of the filter and the character of the data, so that triggers produced come from independent events (noise or signal).

In `lalapps_inspiral` these two steps (thresholding and clustering) were implemented as separate kernels; this optimization fuses them into one. The primary

motivation for this fusion is the thresholding step. Searching through an array for points above threshold is trivial to implement in serial, un-vectorized code. Vectorization or parallelization of this code must be done with care; the problem is equivalent to *stream compaction*, which is difficult to vectorize or parallelize without requiring at least two passes over the array to be compacted [**?**]. However, the number of floating point computations to be performed for each memory operation is very low, and so this kernel will be bandwidth limited; multiple passes over the array incur heavy performance penalties. The primary difficulty is that stream compaction takes its input array and writes out another array consisting of all elements of the input satisfying some criterion, consecutively. This cannot be vectorized or parallelized in one step, because the location to which the output should be written potentially depends on the calculation of all input array elements before any given element.

Fusing the array compaction and the clustering allows us to bypass this difficulty. The key idea is to find the maximum of the output over sub-arrays no longer than the clustering window, and write one output for each such window. We can do this in a single pass over the data, since the output destination is predetermined. We then cluster in a followup pass that looks at the maximum for each window. While that followup pass is not parallelized, in our typical configurations it looks at of order one hundred array elements, rather than a million, and so has trivial cost in comparison. This change greatly improves the performance of both CPU and GPU implementations, and the CPU particularly when multi-threaded FFTs are used to compute the matched filter.

5.5.2 CPU implementation and optimization

We now turn to the specific optimizations and implementation choices necessary for CPU architectures. For concreteness, we focus on the Intel® E5-2670 (Sandy Bridge) product, which is nearly identical (except for slightly lower clock speed) to the cores on Stampede. Our testing included standardized performance tests, employed for all the LSC optimization characterization, with `perf-stat` results given in Section 5.5.2, both below. Similar to Stampede, our reference system has two sockets of eight cores each, running at 2.6 GHz clock speed. All performance results presented here, whether single or multi-threaded, were tested with the CPU affinity of the process set to bind it to a number of cores equal to the number of threads assigned to that process, and

resident on the same CPU socket. CPU throttling and hyper-threading were also disabled for these tests. Each socket has a unified shared L3 cache of 20 MB, and each core has an L1 data cache of 32 KB, and an L2 cache of 256 KB. The architecture supports the AVX (but not AVX2) instruction set, and each core therefore has access to sixteen SIMD registers that can hold either eight single-precision or four double-precision floating point numbers. Potentially one add and one multiply instruction can be retired each clock cycle, so the maximum theoretical peak single precision performance of each socket is $2.6 \times 8 \times 8 \times 2 = 332.8$ GFLOPS. We have tested our code on other CPU architectures as reported in the trade study in Section 5.7; in the following subsections we focus only on the E5-2670. Similar considerations, though with potentially different details, would apply to other CPU architectures that are or might be available to the LSC.

Standard profiling tools can reveal where `pycbc_inspiral` spends most of its time, and timing tests can reveal whether we are in fact able to utilize the most efficient, multi-threaded FFT. Initially, that configuration did *not* give us the highest throughput per socket: the other kernels in the core matched filter were not well parallelized or vectorized and though their cost was small when the program was run in a single-threaded configuration, they became unacceptably slow when the FFT was switched to the faster, multi-threaded configuration. Indeed these kernels before and after the FFT were sufficiently slow in their original implementation that not only did we not achieve close to the matched filter performance expected based on the FFT alone, we did not achieve the highest throughput by running in a multi-threaded configuration. We therefore began our CPU optimization by both vectorizing and parallelizing these kernels, and in the next subsections we report in some detail on those changes, and the resulting performance improvements.

One expensive kernel remains that has not yet received a thorough optimization in its CPU implementation: the time-frequency χ^2 veto. This kernel is more complex and is also only a significant bottleneck when the data quality is poor enough that there are many candidate triggers per segment above threshold. Our next optimization target is a careful vectorization and parallelization of this algorithm. If the autocorrelation χ^2 veto is also shown to be necessary, we will also implement an optimized kernel for the algorithm.

Selecting the optimal FFT library

By design, the count of floating point operations in the basic matched filtering step that compares detector data to a template is dominated by the operation count of the Fast Fourier transform, since that scales as $N \log N$ while other steps scale linearly (or less) in the data segment size. A properly implemented FINDCHIRP executable should therefore likewise have its running time dominated by the FFT, and that FFT should be performed using the most efficient available library.

We have tested two modern, efficient FFT libraries: the Intel Math Kernel Library (MKL) and the Fastest Fourier Transform in the West (FFTW) [**?**]. To make optimal use of these libraries we ensure that all memory provided to FFT calls is 32-byte aligned, and for FFTW that SSE, AVX, and parallelization are enabled within the library. For FINDCHIRP, we must use an out-of-place transform, because the input vector to the FFT (the result of correlating the template with the data) must be preserved in case any points above the runtime-specified threshold are found, as the χ^2 test will require that same input vector.

Finally, because our data analysis pipeline is embarrassingly parallel, there are multiple *a priori* plausible methods of utilizing the multiple cores of the hardware. The legacy `lalapps_inspiral` program was only capable of doing so by running multiple, single-threaded instances, but `pycbc_inspiral` can run multi-threaded. In the standard configuration, either executable performs an inverse, complex-to-complex single-precision out-of-place FFT of length 2^{20}. The input and output to this inverse FFT together require 16 MB of storage, which fits within the L3 cache of a single socket. However, if multiple single-threaded FFTs are performed, they each require this amount of memory but must share the same 20 MB L3 cache; thus, the competing single-threaded processes will be frequently evicting one another from cache and the overall throughput should be expected to decline.

This is indeed what we find; in table 4 we see that eight single-threaded FFTs (for either MKL or FFTW) each require much more than eight times as long as an eight-threaded FFT; the ratio of single-threaded to eight threaded execution time varies from 11 to 20, depending on the transform size and library. But by far the highest throughput for the 2^{20} size FFT is an eight-threaded FFTW implementation, so we wish to design the rest of our executable so that this implementation also retains the highest throughput for `pycbc_inspiral`.

FFT library	Thread configuration	2^{19} length	2^{20} length
MKL	8 single-threaded	10400 ± 130	23500 ± 260
MKL	1 eight-threaded	515 ± 9	2120 ± 60
FFTW	8 single-threaded	7640 ± 510	20700 ± 560
FFTW	1 eight-threaded	432 ± 4	1100 ± 37

Table 4: Time (in μs) to perform an FFT on E5-2670, per invocation. Smaller numbers represent better performance. The 8 thread OpenMP parallel FFTW configuration is the best performing FFT configuration for both transform sizes.

Parallelization of expensive kernels

Both the correlation of the frequency-domain data segment with the frequency domain template (to produce the input to the inverse FFT) and the combined thresholding and clustering algorithm (described in subsection 5.5.1 above, and acting on the output of the inverse FFT) are implemented in the pipeline as C-code kernels. These are parallelized with OpenMP and will dynamically adjust to run on all cores made available to the kernel. The optimal performance was achieved not by a straightforward `for` loop parallelization, but rather by parallelizing a loop that called another function to act on "chunks" of data, where the chunk size is chosen to maximize the amount of data that can fit in the L2 cache of each core.

The quality of parallelization is relatively easy to quantify: a given kernel is benchmarked running on a single core with all other cores idle, and that benchmark compared to the kernel executing on all cores of the socket. Again, we reiterate that we always set the CPU affinity of a kernel so that the operating system cannot dynamically migrate it. If the parallelization is optimal, then the ratio of the single-threaded execution to multi threaded should be the number of cores on the socket, in our case eight.

For correlation of the first half of two arrays of length 2^{20} with output written to a third such array, the parallelized kernel executed on all eight cores in a time of 87.2 μs; the single-threaded kernel in 581 μs, for a ratio of 6.7. For the combined threshold-and-cluster kernel, the eight-threaded kernel executed in 69.3 μs, and the single-threaded in 379 μs, for a ratio of 5.5. While these ratios are not quite at 8, as we would desire, they are still sufficiently close that they do not affect by themselves the performance of the FFT greatly: he difference between the observed multi-threaded performance and the theoretical performance that perfect scaling would imply is of

order 35 μs combined, or roughly 4% of the execution time of the optimal FFT. As described below, other cache effects dominate over this, but when this becomes a bottleneck we will again investigate improving it further.

Vectorization of expensive kernels

The C implementation of the correlation and thresholding has also been vectorized to support SSE4.1 and AVX. The vectorization is hand-coded using compiler provided instrinsic functions that map directly onto SIMD instructions, and the loops are unrolled to permit the vectorized kernel to operate on an entire cache line. Wherever possible memory loads and stores are performed with the "aligned" memory intrinsics, and the arrays on which these kernels act are allocated with 32-byte aligned memory, as they are for the FFT call. Much as for parallelization, for the fused threshold-and-cluster kernel, an efficient vectorization is only possible because of the algorithmic change summarized in section 5.5.1.

As a first estimate of the quality of vectorization, we can benchmark this kernel in isolation and see how many of their instructions are indeed packed AVX instructions; for threshold, this was 99.6%, and for correlate, 100%. Thus the compiler is indeed generating exclusively AVX instructions as we have directed it to via the intrinsic functions. We can quantify the quality of the vectorization similarly to our quantification of the parallelization: benchmarking the kernel with it on and off. In our case it is relatively straightforward to disable most of the vectorization; though it has been hand-coded with vector intrinsics, these are always wrapped in preprocessor directives to allow a graceful fall-back to straight C-code. Hence the intrinsics can be commented out and compiler flags given to prevent the compiler from generating most such instructions on its own[3]. This comparison has been made for both the correlation and thresholding and clustering kernels, where the ratios are 1.83 and 2.34, respectively.

At first sight these ratios appear quite poor, since for the Sandy Bridge AVX instruction set, the peak theoretical speedup from vectorization is a factor of sixteen for single precision code. That factor comes from a factor of eight for the SIMD single-precision vector width and another factor of two because the core can generate

[3]It is not possible to prevent *all* SIMD instructions; because the operating system is 64-bit, the C-library is compiled with a minimal set of SSE instructions, so that turning off all SIMD instructions generates linking errors.

a multiply and an add at each clock cycle. Of course, achieving this peak theoretical speedup is often difficult in practice: the latencies of the multiply and add instructions are five and three clock cycles, respectively, and there are only sixteen SIMD registers that can serve as operands for these instructions. Thus only very specific problems will have the necessary data independence and structure to allow retiring 16 single-precision SIMD arithmetic operations per clock cycle.

Our kernels do not have such structure. The correlate kernel is simpler to analyze, since it is almost identical to element-by-element complex multiplication, for which AVX optimized code is widely available (including from Intel). The only difference between our code and these is that we must add a single instruction, to complex conjugate one of the input vectors. A standard single-precision complex multiplication requires six floating point operations (four multiplications and two additions); an AVX register can hold four single precision complex numbers. Thus the relevant speedup would be how many clock cycles are required to execute the AVX multiplication of the 24 floating point operations equivalent to the multiplication of four complex numbers simultaneously. Because of the need to conjugate an operand as well as the shuffle operations inherent to complex multiplication, there are seven instructions needed for this calculation (there are six in the widely available libraries for AVX complex multiplication; our modification to calculate the complex conjugate adds only a single instruction with a latency of one clock cycle), giving a theoretical speedup of a factor of $2 \times (24/7) = 6.86$, if we were in fact able to retire two AVX instructions per clock cycle. The analysis of the thresholding and clustering algorithm is similar if more complex; each execution of the inner loop requires eight AVX instructions to find the location and values of the maximum of four consecutive complex numbers, which corresponds to 16 scalar floating point operations if we include the comparison. Thus the maximum speedup is only a factor of four, at most.

The further gap between the theoretical peak speedup of vectorization and our measurement can be attributed to memory bandwidth. The correlation kernel reads in two single precision complex numbers—equivalent to four single precision floating point numbers—and writes out a third; between these memory operations, it performs six floating point computations (four multiplies and two adds). There is therefore a one-to-one ratio of memory operations to floating point operations. For the threshold and cluster kernel, two floats are read, and three floating point operations performed,

for a floating point to memory ratio of 1.5. The low floating point to memory ratios mean that any kernel implementing them will be memory bandwidth bound.

We can compare the execution times of these kernels to what memory bandwidth-limited kernels could perform. A correlation for a 2^{20} FFT length must read two vectors of half that length (because the second half is always zero, as part of the FINDCHIRP algorithm to maximize over unknown inspiral phase) and write out a third vector of half that length; a total of 12 MB of memory transactions must occur. If all of that memory lived in the computer's RAM, then we can measure its bandwidth using the STREAM benchmark [**?**]; for a single socket this bandwidth is approximately[4] 26 GB/s. For correlation, this would imply an execution time of 460 μs, much higher than what is measured, and 307 μs for thresholding, again much higher than observed.

That is unsurprising, since we want the data for those calculations to remain in cache and the benchmark performance numbers for those kernels reflect a repeated execution from within cache. Our kernels are parallelized with the goal that each "chunk" remains in L2 cache, which has a published latency of 12 cycles [**?**]. However since our memory for each kernel is accessed sequentially we expect that hardware prefetching ensures that the next data to be read is almost always in the L1D cache, which has a *load* latency of typically five cycles, though it can be as high as seven cycles for AVX loads. For an eight-core E5-2670, which can load or store up to 32 bytes per core, these latencies and the 2.6 GHz clock speed imply an effective load bandwidth of 95 to 133 GB/s. The 87 μs execution of the correlate kernel (which must move 12 MB of memory) would correspond to a bandwidth of 138 GB/s, and the 69 μs execution of the threshold and cluster kernel (which reads 8 MB of memory) would give a bandwidth of 116 GB/s. The correlate kernel slightly outperforms this because its memory accesses are not purely loads. Thus, we conclude that these kernels are bandwidth limited, but achieve essentially the peak bandwidth feasible.

For the two kernels that we have vectorized and parallelized, we find that the parallelization is reasonably good but the performance of vectorization much lower than one might expect. However, this is directly attributable to bandwidth limitation of the kernels, which do achieve close to the peak bandwidth for the architecture.

[4] It is possible to improve this by roughly a third by forcing the use of *streaming stores*; however, while this significantly improves the bandwidth as measured by STREAM, it does so by bypassing the cache on writes. Since the only kernel with significant writes is correlation, this is not beneficial: the output of the correlation *needs* to remain in cache if possible since it will immediately become the input to the FFT.

Performance relative to theoretical peak

We have designed our overall algorithm to be dominated by the FFT, and the optimal FFT implementation to be the multi-threaded FFTW library. Our benchmark above gave approximately 960 μs as the execution time of a 2^{20} single-precision, out-of-place complex inverse FFT; if we use $5N \log N$ as the number of floating point operations performed by the FFT, then this corresponds to a performance of 95 GFlops. For comparison, we also measure the floating point operations using the Linux `perf-stat` tool. That measurement indicated first that 99.999% of the instructions retired were single-precision AVX instructions, so the FFTW library code is extremely well vectorized. The corresponding performance was 91 GFlops, or 83% of the $5N \log N$ estimate. Since there are FFT algorithms with a floating point count as low $4N \log N$, this is consistent with the library having chosen an FFT algorithm with lower floating point cost. With eight AVX capable cores that can retire as many as two AVX instructions per clock cycle, the E5-2670 has a peak theoretical floating-point rate of 333 GFlops; we therefore achieve 27% of the peak flop rate. For an algorithm with the complex memory access pattern of the FFT, this is a not unreasonable performance. Regardless, since we expect to be FFT limited we should not expect higher performance from the `pycbc_inspiral` executable as a whole than this.

The performance of `pycbc_inspiral` depends on the quality of the data. Throughout our benchmarking studies we have consistently followed three different types of data: (i) data which is nearly Gaussian and stationary, representing very good data quality (Type A); (ii) data containing a single, loud transient glitch (Type B), and (iii) data which contains elevated levels of non-Gaussian noise at low frequencies (Type C). The last category is the worst in terms of computational cost, as the χ^2 test must be invoked frequently and the cost is dominated by the computation of that signal-based veto. In late initial LIGO science runs this level of data quality was extremely rare, and should the first observing runs of Advanced LIGO behave similarly, it is not expected to greatly impact the computational cost. The costs we have presented, however, are conservative, and simply average the throughput of the three categories of data.

Measurement of the floating point performance of `pycbc_inspiral` showed 31 GFlops for Type C data, 41 GFlops for loud data, and 44 GFlops for Type A (clean) data. These correspond to fractions of peak theoretical performance of 9.3%, 12.2%, and

Kernel	Type A Data		Type C Data	
	Absolute time (s)	Percentage	Absolute time (s)	Percentage
FFT	1304	60.4	1159	32.3
correlate	332	13.9	300	8.4
template creation	203	9.4	202	5.6
threshold & cluster	97	4.5	87	2.4
χ^2	90	4.2	1601	44.7
data resampling	35	1.6	–	<1
recording triggers	–	<1	49	1.4
Total runtime	2158	100	3583	100

Table 5: Profiling results for Type A and Type C data at a 4096 Hz sample rate on an E5-2670. This table summarized the data shown in detail in Figures **??** and **??** of Appendix5.8.

13.3%. We therefore still have room for improvement, and discuss in the next subsection profiling results and their implications that identify the next priorities for further optimization.

Comparison of measured numbers with theoretical FFT throughput

Finally we assess the overall performance of `pycbc_inspiral` through profiling. Continuing with the same three categories of data, we present a profile run of `pycbc_inspiral` in Table 5 for Type A and Type C data, to illustrate the two extremes, for each kernel costing more than 1% of the overall runtime. From this table, the largest difference we observe is that the χ^2 veto is only 4.2% of the execution time in the Type A data, but 44.7% of the time in the Type C data. This is the reason Type C data is so problematic: in this example χ^2 is calculated four times as often as it was for Type A data. Hence more thorough vectorization and parallelization of this kernel is our next optimization priority.

Since our goal is for the `pycbc_inspiral` engine to be FFT limited, we also use the profile information above to measure the average execution time per FFT *in situ* and compare that to the benchmarked performance for our optimal FFT configurations as shown in Table[5] 4. We present this in Table 6. From these results we see that for the 2048 Hz sample rate, the effective execution time of 516 μs is 84 μs longer than benchmarked average FFT time of 432 μs, whereas for the 4096 Hz sample rate the

[5]Note that the 2^{19} length FFT in Table 4 corresponds to a 2048 Hz sample rate, and a 2^{20} length FFT to a 4096 Hz sample rate.

Sample Rate	Type A Data	Type B Data	Type C data	Average
2048 Hz	517	518	512	516
4096 Hz	1520	1530	1350	1470

Table 6: Effective execution time (μs) of FFT within `pycbc_inspiral` on E5-2670 socket (FFTW, eight-threaded).

observed FFT time of 1470 μs is 370 μs greater than that obtained by benchmarking the FFT in isolation. We can understand this if we recall that the last-level (level 3) cache of the E5-2670 is 20 MB. While the memory of an out-of-place 2^{20} FFT fits inside this at 16 MB, the total memory required for our matched-filter inner loop of correlation, FFT, and threshold and clustering requires a total of 24 MB and does not fit in cache. Because the different areas of memory comprising this 24 MB are accessed at widely separated (in time) parts of this loop, hardware prefetching is unlikely to be able to hide much of this latency. We can validate this explanation by referring to the 2048 Hz sample rate results, where the total memory required by all of the kernels in the matched filter is 12 MB which does fit in cache. And indeed we see that the *in situ* execution time of that FFT is much closer to the isolated benchmark. As a further check, we have counted the number of last-level cache misses of each sample rate, when analyzing the same data with the same bank and number of segments. The 4096 Hz sample rate analysis incurs between 11 and 15 times (depending on data quality) as many cache misses as the 2048 Hz analysis, even though both performed exactly the same number of matched filters.

We are investigating ways to alleviate this penalty, and discuss some of these in the next subsection on future optimizations. Alternatively, it is not yet decided on what hardware the various PyCBC searches will run, and should they do so on hardware with sufficiently large cache the issue could be moot.

Future CPU optimizations

We are investigating a number of performance optimizations to more efficiently implement the existing computational methods: vectorization and parallelization of the template generation and χ^2 veto, and bypassing the CPU cache for loads of some memory, to mitigate the cache eviction causing the degraded *in situ* performance of the 2^{20} size FFT. The latter are in principle possible using the streaming load operations that became available in SSE 4.1, but also require the memory from which they

read to be marked as uncacheable, speculative write-combining (USWC) which is only possible through a kernel module. Aside from these implementation optimizations, as briefly mentioned in section 5.3, we are also exploring alternative scientific methods (such as hierarchical searches and pruned FFTs) that if verified through simulations do not degrade sensitivity can provide potentially large computational savings.

5.6 Justification of Resources

In this section we provide a justification for the total computational cost of the high-latency CBC search, which is summarized in Table 3. All computational cost numbers are quoted in Intel E5-2670 Service Units (SU), which correspond to one core hour on the eight-core E5-2670 chip. The formula used to calculate the computational cost here is substantially the same as that presented in the May 2014 review (c.f. Eq (39) of Section 3.1, page 29 of LIGO-T1400269) with two differences: (i) here we quote search throughput $\mathcal{C}_{\text{throughput}}$ in templates per core per observation hour per detector[6], which is the reciprocal of the quantity used in LIGO-T1400269; (ii) In May 2014, we included the computational cost of the simulations that are needed to tune and measure the pipeline efficiency by reducing the overall template throughput to account for the extra computational cost of the simulations[7]. Consequently, the expression used to compute the computational cost here is

$$\begin{aligned}\text{E5-2670 SUs requested} = (N_{\text{templates}} \times N_{\text{detectors}} \times F_{\text{duty}} \times \mathcal{T}_{\text{observation}}) \times \\ \left[(1 + N^{\text{tuning}}_{\text{injection}}) \times N_{\text{runs}}) + (1 + N^{\text{efficiency}}_{\text{injection}})\right] \Big/ \mathcal{C}_{\text{throughput}},\end{aligned} \tag{5.14}$$

where $N_{\text{templates}}$ is the number of templates (given in Table 7), $N_{\text{detectors}}$ is the number of detectors operating in a given epoch, F_{duty} is the fraction of wall-clock time that the detectors are operating (the detector duty cycle), $\mathcal{T}_{\text{observation}}$ is the total expected duration of observing runs in a given epoch, N_R is an engineering factor that the

[6]We chosen to invert the units since the May 2014 request, so that larger numbers represent better performance.

[7]Including the simulations in the throughput had the effect of reducing the computational throughput of `lalapps_inspiral` from 756 templates per core hour to 198 templates per E5-2670 core hour. The latter throughput number was the basis of our May 2014 cost estimates. When we convert this throughput to Stampede SUs per observation hour per template per detector, this is equal to $1/(198 \times 2.7/2.6) = 4.86 \times 10^{-3}$, as stated in Eq. (32), page 23 of LIGO-T1400269.

accounts for the number of search re-runs needed to account for improvements incorporated in the production search based on our improved instrumental knowledge. Here the new explicit factors

$$N_{\text{injection}}^{\text{tuning}} = 3 \quad \text{and} \quad N_{\text{injection}}^{\text{efficiency}} = 15$$

represent the cost of performing re-analysis of the data with simulated signals for tuning and for final efficiency measurement as a function of binary parameters, respectively. The specific values of 3 and 15 are chosen based on our experience from Initial LIGO searches. A small number of runs suffices to check search efficiency during initial passes through the data, however a large number of injection runs (15) is required to accurately measure search efficiency as a function of parameter space once final calibration, data quality, and tuning information have been incorporated [**?**]. The computational resources requested for injections here is larger than than of the May 2014 request by a factor of 1.6 in O1, 2 in O2, and 2.5 in O3. However, even with the increased cost of injections, our total request is still significantly smaller. In the paragraphs below, we justify the additional factors that enter Eq. (5.14).

Number of Templates $N_{\text{templates}}$: The overall computational cost scales linearly with the number of templates in the search. This depends primarily on three quantities: (i) the boundaries of the astrophysical search space (set by the masses and spins of the target population for a given prioritized science goal); (ii) the desired *minimal match*, which gives the maximum loss in signal-to-noise due to the discreteness of the bank, and (iii) the anticipated shape of the detector's noise spectrum. As in all previous LIGO searches, the bank minimal match is set so that the event rate loss caused by the discrete nature of the bank is less than 10%. This means that the primary drivers of the template bank size are the choice of the boundaries of the astrophysical search space and the anticipated shape of the detector's noise spectrum, as discussed below.

The masses of known NSs are reported to be in the range $0.7M_\odot$ to $2.7M_\odot$ with a mean mass of $\sim 1.4M_\odot$ [**?**], though the lower value, $0.7M_\odot$, comes from an imprecise measurement of a single system that is also consistent with a higher mass. NSs in BNS systems have a more narrow observed mass distribution of $(1.35 \pm 0.14)M_\odot$ [**?**].

The mass distribution of Galactic stellar mass BHs is estimated in [**?**, **?**, **?**], and X-ray observations yield BH masses $5 \leq M_{\bullet}/M_{\odot} \leq 20$, confirmed with dynamical mass measurements for 16 BHs. An apparent lack of BH masses in the range 3–5 $M_{\odot}$ (the "mass gap") [**?**, **?**, **?**] has been ascribed to the supernova explosion mechanism [**?**, **?**]. For the target spin distribution, astrophysical understanding indicates that the older NS in a binary system can be spun up through mass-transfer from its companion, which can increase the spin down time scale. However, this process is not completely understood. The observed dimensionless spins (J/m^2) for NSs in BNS systems (e.g., J0737-3039) are ≤ 0.04 [**?**]. It has been demonstrated that a search for non-spinning BNS systems can capture BNS systems with NS spin up to $J/m^2 = 0.05$ with no loss in event rate [**?**]. We note that the fastest known NS spin is 0.4 [**?**].

Given the current best estimates – and uncertainties – in the masses of compact objects in binaries, we target systems with component masses $m_1, m_2 \geq 1\,M_{\odot}$ and with total mass $m_1 + m_2 \leq 50\,M\odot$. We note that this choice choice of mass space is different from that quoted in the May 2014 review, which assumed a lower mass cutoff of $m_1, m_2 \geq 0.9\,M_{\odot}$ to target more speculative BNS systems. The increase to $1\,M_{\odot}$, which is the result of the scientific prioritization process, results in a template bank that is a factor of ~ 1.3 times smaller than that used for the May 2014 computational cost estimate. The LSC has also prioritized the non-spinning BNS search the highest priority CBC science and a search for spinning BNS as a high priority.

We use X-ray observations of accreting black holes to provide guidance on the expected the black hole spin distributions. Observed black holes spins are distributed over the entire range allowed by general relativity, $0 \leq S/m^2 \leq 1$ [**?**, **?**, **?**, **?**, **?**, **?**, **?**]; both low ($\sim$0.1) [**?**] and high (> 0.85) values [**?**] are represented. Given this uncertainty, the highest priority searches for BBH and NSBH sources must consider BHs with spins the range 0 to 1. Since current searches can only use aligned-spin templates, the bank assumes that the spins of the compact objects are (anti-)aligned with the orbital angular momentum of the binary. An optimal search for system with precession is an (as yet) unsolved physics and data-analysis problem. Simulations will allow us to measure the efficiency and selection biases caused by using an aligned-spin bank to search for an astrophysical population that may contain precessing systems.

Detector sensitivity impacts the computational cost of the CBC search through the detector bandwidth; the computational cost of the search is strongly dependent

on the shape of the detectors' noise spectrum. For the May 2014 review, we calculated the template bank sizes for CBC searches by re-computing the template bank using the zero-detuned high power noise curve (corresponding to aLIGO's ultimate sensitivity for a particular tuning of the detector). We assumed different values of low-frequency cutoffs to model the expected progression in low-frequency sensitivity: $f_{\rm low} = 30$ Hz in 2015/16, $f_{\rm low} = 20$ Hz in 2016/17, and $f_{\rm low} = 10$ Hz in 2017/18. The document "Early aLIGO configurations: example scenarios toward design sensitivity" (LIGO-T1200307) describes plausible scenarios for the strain sensitivity evolution of Advanced LIGO, and now that the Livingston detector has been locked and commissioning has begun, a more informed sensitivity projection can be made, at least for the first observing run. To compute the template bank sizes here, we use the best current estimate of the O1 strain sensitivity, and use the mid-aLIGO and near-final aLIGO curves from LIGO-T1200307 with 20 Hz and 15 Hz low-frequency cutoffs respectively. The exact template bank size will depend on the actual instrumental noise curve, but in the absence of these data we believe that this is a good approximation for bank size, and hence computational cost. If detector commissioning proceeds at a more rapid pace and the zero detuned-high power noise curve is reached in 2017–18, then the computational cost of the search would increase by $\lesssim 50\%$. Table 7 shows the size of the template banks assumed here, measured using the placement algorithm of Ref. [**?**]. We also include the numbers from May 2014 for comparison. The reduction in bank sizes is primarily due to: (i) the factor of 1.3 caused by increasing the minimum component mass of the NS; and (ii) use of a more realistic noise curve than that used in May 2014 to model the detector sensitivity. Since the previously-used zero-detune high power noise curve is significantly "flatter" in the 30–500 Hz region than the predicted noises curve from LIGO-T1200307, it resulted in a bank with a *significantly* greater number of templates. Updated template banks computed using the more accurate noise curve models result in a factor of between 2.7 and 1.1 decrease in computational cost for both the low-latency and offline searches. The largest change is observed in the aligned-spin BNS bank in early runs where the difference between the shape of the zero detune high power curve and the best current prediction is largest. Template banks for a full zero detune high power curve would be 40%–70% larger than the 2017/18 numbers. Future updates to our computing needs will continue to use instrument progress to date and up-to-date predictions for

Signal parameter space	**Number of templates required**		
	2015–16 (O1)	**2016–17 (O2)**	**2017–18**
Non-spinning binary neutron stars	3,780	10,360	23,845
or Aligned spin binary neutron stars	56,440	177,434	486,625
May 2014: Aligned spin binary neutron stars	*196,465*	*435,854*	*1,128,994*
Aligned spin neutron star–black hole binaries	102,163	352,262	1,056,580
May 2014: Aligned spin neutron star–black hole binaries	*213,469*	*559,533*	*2,070,604*
Aligned spin binary black hole search	65,719	239,127	579,971
May 2014: Aligned spin binary black hole search	*106,402*	*242,133*	*631,149*
Total for all CBC searches	**224,322**	**768,823**	**2,123,176**
May 2014: Total for all CBC searches	*516,336*	*1,237,520*	*3,830,747*

Table 7: Number of templates required to cover the different astrophysical targets for CBC searches in each of the three Advanced LIGO observing epochs. The number of templates increases as the low-frequency sensitivity of the detector improves as a consequence of commissioning. The number of templates is measured with the aligned-spin placement algorithm using our current best estimate for the detector's noise curve in each epoch. Note the non-spinning binary neutron star search is a sub-set of the aligned spin binary neutron star search, so only one of these searches will be performed. Shown below each science goal (in italics) is the size of the corresponding May 2014 request from Table 2 of LIGO-T1400269. The reduction in template bank size is due to a factor of 1.3 resulting from the change in the lowest mass neutron star in the bank from 0.9 to 1.0 $M_\odot$, and a factor that varies between 2 (for aligned spin BNS and NSBH in O1) and 1.1 (for BBH in O3) from a more realistic estimate of the detector noise sensitivity.

the sensitivity evolution and the consequences for required computing resources.

Number of Detectors $N_{\mathbf{detectors}}$: In the 2015–16 epoch, only the two LIGO detectors are expected to be operating, so we assume two detectors. In the 2016–18 epoch, LIGO will be joined by Virgo and so we increase the number of detectors to three.

Detector Duty Cycle $F_{\mathbf{detector}}$ and Observation Time $\mathcal{T}_{\mathbf{observation}}$: Given the positive experience with the instrument to date, and the progress made with automated locking scripts, it seems appropriate to plan for the each of the two detectors having an operational availability of 85% for the observing runs. This corresponds to six days per week of operation, with one day for maintenance and commissioning. These activities do not take a full 24 hours, however gaps in observing during the remainder of the week will likely limit duty cycle to the assumed level. The observation time here is based on the advanced detector era run plan of LIGO-T1200307,

which calls for a 3 month run in 2015/16 (O1), a six month run in 2016/17 (O2), and a nine month run in 2017/18 (O3). This is a more conservative schedule than assumed for production analysis in the May 2014 request, which assumed 6 months in 2015/16, 9 months in 2016/17, and 10.8 months in 2017/18. Since CBC search computational cost scales linearly with duration, longer observing times would increase the cost accordingly.

Engineering Factor for Number of Re-runs $N_{\mathbf{runs}}$: Experience with Initial LIGO has shown that a significant number of pipeline re-runs may be needed the first time we search with an unknown detector. In the first epoch (2015–16) we expect the number of re-runs to be four to account for improvements in tuning, data characterization, calibration, and addition of new algorithms (e.g. improved signal-base vetoes) into the pipeline. Experience has also shown that it is necessary to re-analyze the entire observing run run to account for these factors, as the quality of the detector data can very significantly during a run. As our experience with the Advanced LIGO data increases, we expect the engineering factor to decrease to 2 in 2016–17, and finally to 1 in 2017–18. We note for S6/VSR2,3 data (the final observing run of Initial LIGO/Virgo) four re-runs were needed to obtain the final result.

Computational Throughput $\mathcal{C}_{\mathbf{throughput}}$: We used the measured computational throughput of the best optimized `pycbc_inspiral` implementation on the Intel E5-2670 (as described in Section 5.5.2) to compute the overall computational cost. As described above, the run-time of the code is not deterministic; the number of operations needed depends on the features in the LIGO detector's noise background. To account for this, we benchmark the code on three types of data (labeled A, B, and C) from Initial LIGO, which are representative of different types of data quality: (i) a clean stretch of data (Type A); (ii) a stretch containing a single loud transient glitch (Type B); and (iii) a stretch with elevated levels of non-Gaussian noise at low frequencies (Type C). Type C is the worst type of data quality in terms of computational cost. The speed of the analysis is significantly slower for this type of data, as more time is spent computing signal based vetoes. To compute the computational cost of the analysis, we take the mean of the three measured throughput values. If the data is cleaner (meaning that there are less non-Gaussian transients) the throughput could be closer to the higher numbers. If the data quality is poor, throughput could be closer to the lower numbers. Investigation of early Advanced LIGO data from the

Livingston detector indicates that this is a reasonably conservative estimate for the (as yet unknown) quality of data in the first Advanced LIGO observing run. The throughput numbers measured in the three data types at a sample rate of 4096 Hz, are:

$$\mathcal{C}^{\mathrm{A}}_{\mathrm{throughput}} = 6390 \text{ templates/core}, \tag{5.15}$$

$$\mathcal{C}^{\mathrm{B}}_{\mathrm{throughput}} = 5887 \text{ templates/core, and} \tag{5.16}$$

$$\mathcal{C}^{\mathrm{C}}_{\mathrm{throughput}} = 3673 \text{ templates/core} \tag{5.17}$$

giving an average throughput of $\mathcal{C}_{\mathrm{throughput}} = 5316$ templates/core. The template throughput numbers measured for `pycbc_inspiral` are a factor of 7 better than the best performance observed by `lalapps_inspiral` on the Intel E5-2670 due to the algorithmic, library, and code optimizations described in Section 5.5.

Using the factors above, we compute the computational cost required for the production high-latency CBC computing in Millions of Service Units (MSU), shown in Table 3. The costs of the aligned-spin searches in this table reflect our best knowledge of the computational cost, given the factors discussed above. Since these numbers were derived by benchmarking the Advanced LIGO search code, and our experience analyzing data from the six Initial LIGO science runs, they represent our best estimate of the cost of the CBC searches in Advanced LIGO, given the uncertainties in data quality, duty cycle, and detector bandwidth.

Reducing the sample rate by a factor of two to 2048 Hz increases the throughput to $15,025$ templates/core, $13,530$ templates/core, and 7024 templates/core for the three data types A, B, and C, respectively. The larger increase for types A and B is due to better use of Level 3 cache in the FFTs, which dominate for the cleaner data. It has not yet been demonstrated that the sample rate for the deep, offline search can be reduced and so we base our estimates on the sample rate used in Initial LIGO. If this reduction in sample rate can be achieved, either by hierarchical methods or sub-sample interpolation, it represents another possible optimization, as discussed in Section 5.3.

5.7 Selecting optimal hardware solutions

One of the primary factors guiding the design of the new PyCBC framework was to provide the ability to implement compute kernels on a variety of architectures, including CPUs, GPUs, and the Intel Many Integrated Core Architecture (MIC) co-processors. Over the past year, we have focused on implementing the FINDCHIRP algorithm on GPUs, as the NVIDIA CUDA FFT library [**?**] implements an extremely efficient "black box" FFT that scales very well to the large 2^{20} (and longer) complex FFTs used in the offline CBC search. Furthermore, since matched filtering LIGO data can be performed at single precision, we have investigated inexpensive consumer-grade GPU cards as a possible computing platform for the high-latency CBC search[8]. Section 5.7.1 describes the CUDA implementation of the FINDCHIRP algorithm and the initial results of our GPU hardware trade study. Finally Section 5.7.2 describes the results of our trade study investigating the performance of the best CPU `pycbc_inspiral` implementation on different CPUs, including Intel Westmere, Sandybridge, Ivybridge, and Haswell.

5.7.1 PyCBC on Graphics Processing Units

Our goal when implementing the GPU-enabled version of PYCBC_INSPIRAL is to execute as much computation on the GPU, with as little data passing over the (slow) PCIe host interconnect as possible. Simply off-loading the FFT to the GPU does not significantly speed up the code, due to the rate-limiting step of moving the input and output vectors over the PCIe bus. Fortunately, the FINDCHIRP algorithm lends itself well to performing all computations on the GPU, as the pre-conditioned input data segments can be stored in global GPU memory and then processed through many templates that are generated on the GPU. Our GPU implementation therefore implements as CUDA-native kernels *both* the compute-intensive steps of the algorithm (correlate, FFT, and time-frequency signal-based veto) *and* the relatively light-weight steps (template generation and threshold/cluster), ensuring that only very minimal PCIe bandwidth is required to initially stage the data to GPU memory and pass triggers back to host memory.

For large regions of parameter space, template generation can be expressed as an

[8]NVIDIA artificially reduces the speed of double precision arithmetic on the consumer-grade GPU units, but single-precision arithmetic runs at full speed.

analytic polynomial, which we have implemented as a straightforward element-wise GPU kernel. Work is ongoing on extending template generation to other waveform approximants that are more appropriate for modeling higher mass BBH systems. As the correlate kernel is a point-wise complex multiply and conjugate, the GPU implementation is also straightforward. We make use of NVIDIA's proprietary cuFFT library to perform inverse FFTs. This library factors the FFT into multiple kernel calls based on the size of the FFT and the GPU hardware capability. On a Tesla K10, using CUDA 6.5, FFT sizes between 2^{20} and 2^{23} all factor into three kernels calls. As the FFT is memory bandwidth bound, it is clear that for these range of sizes the FFT throughput will scale linearly with vector length. Thresholding and clustering is divided into two kernels. The first performs both thresholding and local peak finding on small fixed window sizes. The kernel window sizes are smaller than

GPU Card Type	Memory Bandwidth (GB/s)		SP Performance (GFLOPS)		FFT GFLOPS	Cost
	Theoretical	Measured	Theoretical	Measured	Measured	
GTX 580	192	170	1581	1553	444	N/A
GTX 980	224	179	4612	4980	456	$555
GTX 970	224	155	3494	4025	357	$329
GTX 750 Ti	86	80	1306	1490	150	$139
Tesla 2090	177	106	1331	1309	361	N/A
Tesla K10	160	101	2290	2015	288	$2800
Tesla K80	240	170	4350	3712	288	$5000

Table 8: Theoretical and measured performance of the GPUs investigated in our trade study. The theoretical performance for the consumer grade cards is taken from the NVIDIA reference implementation of the GPU. Faster than theoretical measured performance can be obtained if the consumer card manufacturers (e.g. PNY or EVGA) overclock their cards compared to the reference implementation. Columns two and three compare the published theoretical and measured bandwidth from the GPU global memory to the processor (in GB/s). Columns four and five compare the published theoretical single-precision computational speed (in GFLOPS) for the reference implementation of the GPU with the speed measured on our cards. Column six shows the computational speed of the cuFFT for large transforms, which is memory bandwidth limited and column seven shows the (March 2015) cost of the card. We use ORNL SHOC to measure the *in situ* performance. Note that the Tesla K10 and K80 GPU boards contain two independent GK104 and GK210 GPU chips. The performance numbers quoted here are for a single GPU chip, and not the board. The GTX 580 and Tesla M2090 are no longer in production. These cards cost $500 and $2500, respectively, when purchased.

the scientifically chosen clustering window. This exposes an additional parallelism. A second, very short-running kernel that executes a single block, is used to perform final cleanup and boundary condition checking. Following this kernel, we dump triggers back to the host, which due to the on-GPU clustering is guaranteed to be $O(10^{-3})$ the size of the data vectors in the worst case, and on average much less. Finally, we have also implemented our time-frequency signal consistency test as a set of GPU kernels where each is designed to handle a different number of triggers. This is implemented using a standard parallel reduction sum operation.

GPU Benchmarking Results

Similar to the CPU implementation, the 3 kernels that dominate the inner loop of the matched-filter (correlate, FFT, and thresholding) are all memory bandwidth bound. Therefore both memory bandwidth and floating point performance are considerations when selecting the optimal GPU hardware. Table 8 shows the GPU hardware that we have procured to test the CUDA implementation of the FINDCHIRP algorithm[9]. We have benchmarked `pycbc_inspiral` on these GPUs, with the results shown in Table 9 in templates per GPU chip (note that in practice, the overall throughput of the K10 is a factor of two faster than the numbers quoted here, as it contains two GPU chips per PCIe board).

Using templates per dollar as the performance benchmark, the best performing GPU is the GTX 750 Ti, with an average throughput of 790 templates per dollar. This is the cheapest of the consumer-grade GPUs that we have tested. This card also has the advantage that it is powered by the PCIe bus (no additional 6- or 8-pin PCIe power connectors are needed) and it can easily be converted to a 1U profile. For comparison, the Intel E5-2670 (which currently retails for $1365) has an average throughput of 42,500 templates per socket or 31 templates per dollar. The best performing CPU we have tested using the cost metric is the Intel E3-1220-v3 which has a throughput of 28,340 templates per socket at a cost of $205, or 138 templates per dollar. It is important to note that this metric oversimplifies the comparison between CPUs and GPUs, as GPUs require a CPU-based host system. The true cost metric should take this into account. However, the consumer-grade cards are a very promising avenue for exploration for co-processors in CPU-based systems or in

[9]Unfortunately, our Tesla K80 proved to be unstable in the Super Micro test chassis that we are using, and so we have not been able to measure sustained performance on this card.

GPU Card Type	Search throughput (templates / GPU board)			Average
	Type A Data	Type B Data	Type C Data	throughput
GTX 580	216,100	210,000	151,700	192,600
GTX 980	221,000	213,800	154,900	196,600
GTX 970	199,300	194,000	144,800	179,400
GTX 750 Ti	120,700	116,600	92,000	109,800
Tesla 2090	154,800	149,000	117,200	140,300
Tesla K10	133,400	126,600	95,200	118,400

Table 9: Computational throughput of `pycbc_inspiral` running the CUDA GPU compute kernels on consumer and HPC-grade GPUs for the three types of data used for benchmarking. The input data sample rate is 4096 Hz and the code processed 15 data segments of length 256 s through each template (for comparison with the CPU numbers). Two independent `pycbc_inspiral` processes schedule CUDA kernels on a single GPU chip to maximize the throughput. For comparison, the throughput of an eight core E5-2670 socket is 51,120 templates/socket for Type A data and 29,400 templates/socket for Type C data.

custom GPU clusters. We can also use the CUDA kernels in `pycbc_inspiral` to take advantage of XSEDE resources that provision HPC-grade GPUs.

Consumer-grade GPU units do not have the memory error checking and correction (ECC) provided in the NVIDIA's HPC GPU line[10] and so a concern when constructing consumer-grade GPU clusters is reliability of the compute units. To investigate the reliability of the consumer grade cards, we run the ORNL SHOC `stability` code [**?**]. This program generates a series of random numbers that and then repeatedly computes the forward and reverse FFT of the input vector. On each iteration, the code checks that the calculated output vector agrees with the original input vector and reports errors. A sustained two-day test with 13 GTX750 Ti cards showed no errors. This is more encouraging than our original tests with GTX580 cards, which showed a relatively high rate of errors[11]. We attribute this difference to the fact that the GTX 750 Ti runs at lower clock speeds and lower power than the GTX580. Testing of the GeForce 900 series of cards to see if they also show this level of reliability is ongoing. Given these considerations, consumer grade GPU cards can yield a very cost-effective platform for the offline CBC search.

[10] We note that Tesla ECC is implements in software, which further slows down the throughput of the cards from theoretical peak.

[11] However, since the GTX580 was a factor of five less expensive than the equivalent Tesla M2090, it was still more cost-effective, even at the cost of running all computations twice to check for errors.

Optimization of the GPU Implementation

While our initial CUDA implementation of the the FINDCHIRP algorithm is efficient in the sense that as much computation is performed on the GPU as possible, we have identified several areas for future optimization. Several of these optimizations are in progress, but others require assistance from the NVIDIA CUDA and cuFFT engineers as they require re-design of the cuFFT API. To achieve these changes, we have established a collaboration with Mike Clark, an NVIDIA engineer resident at Caltech, and Alex Fit-Florea, head of the NVIDIA cuFFT development team. We describe our planned optimizations below.

Since all of our input data is staged to the GPU, the rate limiting factor for our current implementation is the memory bandwidth between the GPU's global memory and the on-chip Level 2 cache and registers where threads access data for computation. Our primary goal in optimizing the GPU implementation has been to reduce the number of memory transfers and maximize the use of the GPU's floating point engine. CUDA kernels operate on data in GPU global memory and for each kernel call, data is transferred across the memory bus[12] from GPU global memory to the registers of the processor cores and back to global memory at the end of the kernel. A basic performance analysis can be obtained by counting the memory operations executed by the correlate, FFT, and threshold kernels used in the FINDCHIRP loop:

$$\text{Correlate}(2\text{in} + 1\text{out}) + \text{FFT}(3\text{in} + 3\text{out}) + \text{threshold}(1\text{in}) = 10 \text{ memory transfers} \tag{5.18}$$

With the release of CUDA 6.5, a new feature was added to the cuFFT library that allows user defined callback functions for both the load of the initial input vector and the store of the final output vector of the FFT. This has the potential of allowing us to fuse computations from the correlate and threshold steps into the FFT kernel, reducing the number of memory transfers and increasing performance. Our first step towards optimizing our CUDA implementation has been to investigate the use of callbacks.

The current implementation of NVIDIA's cuFFT callback API allows element-by-element functions to be easily applied, with no guarantee about the relationship

[12] Typically DDR3 or GDDR5 depending on the model of GPU card.

between nearby elements or order of operations within the kernel itself. Because the callbacks cannot be compiled into the FFT kernels themselves, and can handle only single elements, there is significant overhead to their use that cannot be easily predicted without benchmarking. Fig. **??** compares the relative execution time of the three kernels that make up the inner loop of the matched-filter code under three cases. The first case (left) uses the initial kernel implementations without making use of the callback API. The second case (middle) fuses the correlate kernel, without modification, into a load callback. We see that there is a noticeable drop in the total execution time. The savings comes from the removal of both a full vector length store and read operation. Note however, that this is significantly less improvement than would expected from a naive counting of the memory savings. The final case takes full advantage of the known contiguous regions where the input vectors are zero, and where the output vector does not produce valid results due to wrap-around corruption. Callbacks appear to be a very promising avenue of optimization, and our collaborators on the NVIDIA cuFFT team are interested in our application as a use-case for developing the API further.

For certain kinds of commonly used waveform templates, in particular the TaylorF2 approximant, the amplitude of the waveform is a simple power series. This allows it to be precomputed, and instead of including it with the template itself can be pre-multiplied into the segment of data to analyze. Where this is possible, the remaining portion of the template can be expressed in the form $e^{i\psi(f)}$. It is possible to trade floating point operations for a savings in global memory reads by storing only the Fourier phase of the template, $\psi(f)$, and recalculating the full $e^{i\psi(f)}$ within a load callback of the FFT. If the callback API can be extended to allow a vectorized version of the store callback that operates on contiguous elements, it may be possible to merge a portion of the peak finding algorithm into the store callback, vastly decreasing the memory writes at the end of the fused kernel.

More optimal use of the available memory bandwidth can also be achieved by reducing the amount of data sent over the memory bus. We are investigating the possibility of storing the output SNR time and input template phase as half-precision (FP16) numbers to reduce memory bandwidth. We have also discussed with NVIDIA the possibility of adding callbacks to the intermediate steps of the cuFFT implementation (since our 2^{20} point FFTs are implemented by three kernel calls in cuFFT)

that would allow us to use FP16 precision between each FFT radix. Performing the FFT operations in FP32 and storing the intermediate products in FP16 may be possible. We are beginning a study to determine if this model could meet our accuracy requirements.

Finally, we are investigating the optimal GPU/CPU ratio for systems and parellization between the host CPU and GPU kernel execution. As GPU kernel launches are asynchronous compared to host execution, it is possible to hide trivial serial operations that occur within the host code. The exception is where triggers are offloaded from the GPU onto the CPU, which is a blocking operation. Host execution does not proceed until the GPU queue is drained. When the data is synchronized there is a noticeable delay before new GPU kernels are executing. This can be minimized by executing multiple host processes that submit work to the same GPU, and by batching additional work together to amortize the device offload latency. We have shown that two processes running on the same CPU launching kernels to a single GPU makes more efficient use of the GPU resources; tests to find the optimal ratio are ongoing.

5.7.2 CPU Hardware Trade Study

We have benchmarked the current best implementation of the `pycbc_inspiral` executable several different CPU systems and studies are ongoing to determine the most cost-effective CPU configuration for the high-latency CBC search. Benchmarking was performed on a dedicated machine, with all cores occupied with the CBC search code to simulate production use (either in single-thread or multi-thread mode). For FFTW benchmarking, plans were measured using the patient measure level for the appropriate hardware. Table 10 shows the benchmarking results for the Intel Westmere, Sandybridge, Ivybridge, and Haswell CPUs that we have tested to date. Similar to the low-latency results, the fastest throughput is obtained on the Intel E5-1660-v3 Haswell processor. One interesting result we note is that the fastest throughput per *active* core is obtained on a 10 core Ivybridge E5-2670 v2 CPU when FFTW is run with two cores disabled. This also yields a slightly faster throughput per socket, even when we account for the fact that we are paying for two disabled cores. We attribute this to the fact that FFTW achieves better performance with power-of-two numbers of cores, and the E5-2670 v2 has a 25 MB level 3 cache, and so at the 4096 Hz sample

rate, almost all of the data (correlation and FFT) fits into cache. We are continuing to explore different CPU clock speed, cache size, and cores per socket configurations to determine the best throughput, although this also involves writing PyCBC CPU kernels specific to e.g. AVX2 instructions to obtain best performance. These optimization efforts are ongoing as part of our software improvements and hardware trade study.

5.8 Comparison of LALApps and PyCBC Profiling

In this appendix we provide complete profiling for the `lalapps_inspiral` executable used to estimate computational cost for May 2014 review and the new `pycbc_inspiral` executable used in the current estimate. The benchmarked speed for `lalapps_inspiral` quoted in the May 2014 review was 756 templates per E5-2670 core. We have re-benchmarked the `lalapps_inspiral` code to confirm these numbers and obtain, when averaging over the three data types A, B, and C, a throughput of 788 templates per E5-2670 core, consistent with the performance measured in April 2014. Table 11 shows the complete call graph for one invocation of the retired LALApps executable. Using this information, we identified the time-frequency signal-based and the FFT engine as the first targets for optimization. We used additional profiling to identify the non-FFT parts of the code (e.g. `LALFindChirpClusterEvents` which performs clustering and `LALFindChirpFilterSegment` which performs correlation and thresholding) that were causing performance bottlenecks in the use of parallel FFTs. For comparison, we show the current `pycbc_inspiral` executable run on the same data and template bank, using eight-thread FFTW. The improvements that we have made cause the total execution time to drop so significantly (from 3556 seconds to 140 seconds) that the profile information is dominated by the data reading and pre-conditioning, as shown in Table 12. To ensure that the matched filtering dominates the profiling and throughput measurement of `pycbc_inspiral`, we have run all other `pycbc_inspiral` tests on significantly larger template banks than the `lalapps_inspiral` executable was able to process (57,222 templates per bank compared to 2469 in May 2014).

Finally, Figures **??** and **??** show the full call profile graphs for the `pycbc_inspiral` executable on Type A (clean) data and Type C (non-Gaussian) data respectively. These call graphs show profiling information for the runs used to produce the template throughput measurements on our reference CPUs given in Eqs. (5.15) and (5.17). The

Processor	Base clock speed (GHz)	cores / socket	Level 3 Cache (MB)	LIGO Data sample rate (Hz)	Search Throughput (templates/active core)		FFT Engine
					Type A Data	Type C Data	
E5-1660 v3	3.0	8	20 MB	4096	9603	7395	FFTW 8 thre
E5-2670 v2	2.5	10	25 MB	4096	8211	4398	FFTW 8 thre
E5-2640 v3	2.6	8	20 MB	4096	8131	6162	FFTW 8 thre
E3-1220 v3	3.1	4	8 MB	4096	7402	6579	MKL 1 thread
E3-1241 v3	3.5	4	8 MB	4096	7351	6225	FFTW 4 thre
E3-1220 v3	3.1	4	8 MB	4096	6842	5497	FFTW 4 thre
E5-2670 v2	2.5	10	25 MB	4096	6395	3523	FFTW 10 thr
E5-2670	2.6	8	20 MB	4096	6390	3673	FFTW 8 thre
E5-2670	2.6	8	20 MB	4096	5878	3606	MKL 1 thread
X5650	2.66	6	12 MB	4096	4320	3264	MKL 1 thread
E3-1220 v3	3.1	4	8 MB	2048	17952	12394	FFTW 4 thre
E3-1220 v3	3.1	4	8 MB	2048	15226	11834	MKL 1 thread
E5-2670	2.6	8	20 MB	2048	15025	6508	FFTW 8 thre
E5-2670	2.6	8	20 MB	2048	10414	6306	MKL 1 thread
X5650	2.66	6	12 MB	2048	8230	5472	MKL 1 thread

Table 10: Comparison of `pycbc_inspiral` throughput on different Intel Westmere (X5650), Sandybridge (E5-2670), Ivybridge (E5-2670 v2), and Haswell (E3-1220 v3, E5-1660 v3, E5-2640 v3, and E3-1241 v3) CPUs. Search throughput is given in templates per active socket used in the FFT. We also show the throughput for two different input data sample rates, 4096 Hz (as used in the S6/VSR2,3 analysis) and 2048 Hz, to illustrate the effect of Level 3 cache size on the throughput. We note that the fastest throughput per active socket is from the 10-core Ivybridge processor with eight-core multi-threaded FFTW. We attribute this to the increased size of the cache (25 MB) and the effect of non-power-of-two transforms. For a cost metric, we should compute the throughput per socket, however even with this metric the E5-2670 v2 CPU is faster when run in 8-core mode than in 10-core mode (65,680 vs 63,950 E5-2670 v2 templates per socket). We also note that for the cheaper E3-1220 v3 Haswell with an 8 MB Level 3 cache, single-threaded MKL yields the fastest throughput. We also note that the code used has been optimized to the Sandybridge architecture, and further optimization to the Haswell architecture may be possible. These comparisons illustrate the type of considerations that we are exploring to determine the most cost effective hardware.

code is run on a 57,222 template bank so that the matched filtering dominates the run-time of the executable. Each box in the call graph is labeled with a Python module name, line number, and function call on the first line. The second line shows the cumulative percentage and, in square brackets, the absolute time in seconds spent

in that function *and all of its children*. The third line shows the percentage of time actually spent in that function as a percentage and an absolute time in seconds. The fourth line shows how many times that function was called. Note that when reading these call graphs, the `scipy.weave` compiled code (including the compiler intrinsics) appears in a single function called `~:0:<apply>`, so it is necessary to look at the parent function of this call to determine how much time is spent in a higher-level operation. The highest-level user-code function is the third box labeled `pycbc_inspiral:19:<module>`. Clearly, 100% of the time will spent in that function

```
CPU: Intel Sandy Bridge microarchitecture, speed 2600.04 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Clock cycles when not halted) with a unit mask of 0x00 (No unit mask) count 100000
%        total time   image name               symbol name
31.179   2084.289   liblalinspiral.so.9.0.0    Chisq_CPU
16.502   1103.165            libmkl_avx.so     mkl_dft_avx_ipps_cFft_BlkSplit_32fc
11.120    743.379            libmkl_avx.so     mkl_dft_avx_ipps_cFft_BlkMerge_32fc
5.896    394.141             libc-2.12.so     __GI_memset
5.801    387.804            libmkl_avx.so     anonymous
4.232    282.916            libmkl_avx.so     mkl_dft_avx_ipps_cFftInv_Fact4_32fc
4.139    276.692            libmkl_avx.so     anonymous
4.027    269.185            libmkl_avx.so     anonymous
3.987    266.558            libmkl_avx.so     anonymous
2.329    155.694   liblalinspiral.so.9.0.0    XLALBankVetoCCMat
1.831    122.402            libmkl_avx.so     anonymous
1.771    118.371             libm-2.12.so     __ieee754_log
1.428     95.462            libmkl_avx.so     anonymous
1.396     93.296            libmkl_avx.so     anonymous
1.268     84.799   liblalinspiral.so.9.0.0    LALFindChirpClusterEvents
0.677     45.257              no-vmlinux      /no-vmlinux
0.663     44.315   liblalinspiral.so.9.0.0    LALFindChirpFilterSegment
0.471     31.460   liblalinspiral.so.9.0.0    LALFindChirpSPTemplate
0.277     18.511             libc-2.12.so     memcpy
0.128      8.543        libFrame.so.1.4.1     Frz_inflate_fast
0.124      8.303             libm-2.12.so     log
0.121      8.102            libmkl_avx.so     mkl_dft_avx_ipps_cFftInv_Large_32fc
0.105      7.033            libmkl_avx.so     anonymous
0.081      5.395          liblal.so.9.0.0     XLALIIRFilterREAL4Vector
0.072      4.827        libFrame.so.1.4.1     FrCksumGnu
0.049      3.249          liblal.so.9.0.0     XLALREAL4ReverseFFT
0.037      2.460            libmkl_avx.so     workaround_for_DFTInv_RPack_32f
0.033      2.173             libm-2.12.so     isnan
0.026      1.718   liblalinspiral.so.9.0.0    LALFindChirpComputeChisqBins
0.021      1.417          liblal.so.9.0.0     XLALIIRFilterREAL8Vector
0.021      1.417          liblal.so.9.0.0     XLALIIRFilterReverseREAL8Vector
0.021      1.397            libmkl_avx.so     mkl_dft_avx_ipps_cCcsRecombine_32f
```

Table 11: Complete profiling information for the fastest configuration of the retired `lalapps_inspiral` executable used for cost estimates in May 2014. The code uses the single-threaded Intel MKL FFT engine and is run on an Intel E5-2670. The total run-time of the code is 6685 seconds, with a total of 3556 seconds spent in the MKL FFT routines. The next largest consumer of CPU time is the `Chisq_CPU` function that calculates the time-frequency signal-based veto. Improvement of this algorithm, as described in Section 5.5.1 was our first priority for optimization.

and its children, since it is the main program. However, only 1.39% of the total execution time is spent actually in that module. In the Type A (clean) data shown in Figure **??** the FFT engine dominates the computation cost, with the correlation used to compute the integrand of the matched filter the second dominant function at 15% of the run-time. In the Type C data, significantly more time is spent computing the time-frequency χ^2 signal-based veto; this is our next target for optimization.

5.9 Development and Simulation Costs

The May 2014 request for computational resources presented in LIGO-T1400269 presented the total production computing request, but it did not include an estimate of the computational resources needed to develop, test, and tune the searches prior to the observing runs. For completeness in this appendix, we present our best estimate of the resources needed to develop and test the high-latency CBC search before the observing runs, using the measured throughput for the best optimized `pycbc_inspiral`. The request in MSU is for all high-latency CBC computing summarized in Table 13. Resources needed for development are larger relative to production in the earlier epochs. This is due to the fact that more time is spent in engineering runs, mock data challenges, and development and reduce in later years as more time is spent in observing runs. The requested time for simulations includes the computational resources necessary to review the analyses prior to observing runs.

To obtain development costs, we assume that the detectors will be operating with

```
ncalls  total time  filename:lineno(function)
37035   55.188      fftw.py:451(execute)
92301   32.677      {apply}
1       22.851      {scipy.signal.sigtools._linear_filter}
1        8.896      {_lalframe.FrStreamReadREAL8TimeSeries}
1        2.506      {_lal.HighPassREAL8TimeSeries}
```

Table 12: The `pycbc_inspiral` executable run on the same input data and template bank as shown for `lalapps_inspiral` in Table 11. The PyCBC code is so much faster that the data reading and conditioning are a significant fraction of the run-time. To ensure that the matched filtering dominates the profiling of `pycbc_inspiral` we run on significantly larger template banks than the `lalapps_inspiral` executable was able to process, as shown in Figures **??** and **??**.

a 20% duty cycle between observing runs. This time will comprise engineering runs as well as night and weekend running when the detector is in a sufficiently stable state during commissioning to record data. We assume that only three sets of tuning injections are performed. The simulation request is based on experience with the number of mock data challenges needed to prepare for O1: we expect to analyze approximately 3 months of data between O1 and O2, and 1.5 months of data between O2 and O3 in mock data challenges. However, the actual costs will be updated based on experience.

Astrophysical search target	**E5-2670 MSU per**	
	2015–16	**2016–17**
Production: Binary neutron stars (non-spinning)	0.084	0.514
Production: Binary neutron stars (aligned-spin)	1.25	8.82
Development and Simulation: Binary neutron stars (non-spinning)	0.169	0.218
Development and Simulation: Binary neutron stars (aligned-spin)	2.49	3.70
Production: Neutron star–black hole (aligned spin)	2.26	17.5
Development and Simulation: Neutron star–black hole (aligned spin)	4.52	7.32
Production: Binary black hole search (aligned spin)	1.45	11.9
Development and Simulation: Binary black hole search (aligned spin)	1.54	3.28
Total for high-latency CBC development, simulation, and production	**13.8**	**53.3**

Table 13: The total computational resources needed for production, simulations, and development for the high-latency CBC search in millions of service units (MSU) per year. One service unit is defined as one core hour on an Intel® E5-2670. The production request is identical to the request in Table 3 in this document. The development and simulation request are our best estimates of the computational resources required to develop and tune the searches prior to the observing runs in each epoch. The total for production, simulations, and development is reflected in the total CBC request in the LSC Computing Plan LIGO-T1500118 (see e.g. Table 3, page 13 of LIGO-T1500118 for comparison with the total O3 request).

Chapter 6

Focused BNS Analysis

6.1 Introduction

The coalescence of compact binary systems are a promising source of gravitaional-wave detections from the next generation observatories, Advanced LIGO (aLIGO) and Advanced Virgo (AdV). Binary neutron star systems are likely to be one of the first sources observed by these observatories. Advanced LIGO will begin its first observing run (O1) in the fall of 2015, and will reach design sensitivity by 2018-19. Detection rate estimates for aLIGO and AdV sugest that BNS sources will be one of the most numerous source detected, with plausible rates of $\sim 10/\mathrm{yr}$ [**?**]. The detection of multiple BNS systems will allow us to explore the processes of stellar evolution and measure the properties of a nuetron star, including information about the nuclear equation of state[**?**].

Current electromagnetic observations suggest that the neutron star mass distribution peaks at $1.35M_{\odot}$–$1.5M_{\odot}$ with a narrow width [**?**], although neutron stars in globular clusters seem to have a considerably wider mass distribution [**?**]. There is also evidence that a neutron star in one system has a mass as high as $\sim 3M_{\odot}$ [**?**]. The dimensionless spin magnitude $\chi = cJ/Gm^2$ for neutron stars is constrained by possible neutron star equations of state to a maximum of 0.7 [**?**]. The fastest observed pulsar has a spin period of 1.4 ms [**?**], corresponding to a $\chi \sim 0.4$, and the most rapidly spinning observed neutron star in a binary, J0737–3039A, has a spin of only $\chi \sim 0.05$.

The gravitional wave from an inspiralling binary system can be separated into

an inspiral portion, where the wave is slowly increasing in amplitude and frequency, a merger, and the post-merger signal. For systems with a total mass is less than $\sim 12M_{\odot}$, and where the angular momenta of the compact objects is low, as is the case with BNS systems, it has been shown that post-Newtonian approximants, which model only the inspiral portion of the waveform, and are currently available at up to 3.5PN order, can provide an accurate model of the gravitational waveform for the purposes of detection.

Since we have a well-modelled gravitational wave signal, searches for gravitational waves from CBC sources use template-based matched filtering [**?**]. The data from a detector is correlated against of bank of known template waveforms. The set of templates is chosen so that any signal will lose no more than 3% of the optimal signal-to-noise ratio that would be obtained by an exactly matching template in the absence of noise. Geometric metric-based methods for placing both non-spinning and aligned spin templates have been shown to be effective for binary neutron stars [**?**].

In addition to possible signals, and Gaussian noise, detector data contains non-Gaussian noise transients, which can generate large, spurious SNR triggers. To mitigate the effect of these noise transients, signal-consistency tests are used to create a weighted detection statistic. In addition, a signal must been seen in multiple detectors with consistent parameters. It has been shown that using a bank of templates shared between all detectors, and requiring a signal to be observed in the same template accross the detector network, improves the overall search sensitivity [**?**].

In this paper we present an offline search pipeline tuned for the detection of BNS sources, and show that this targeted search yields significant improvements in sensitivity to BNS sources. Whereas in prior searches for BNS systems, such as the last one conducted in S6/VSR2,3, a nonspinning template bank was constructed that contained masses up to $25M_{\odot}$ [**?**] and was tuned to be sensitive in all regions, we focus on soly on BNS systems with a mass range from $1 - 3M_{\odot}$. To approximate the conditions of the first observing run with Advanced LIGO, we focus on a two-detector network composed of the Hanford (LHO) and Livingston (LLO) observatories.

This paper is organized as follows. In sec. 6.2 we describe the methodology of the search pipeline, and we present a method for estimating the significance of candidate events. In sec. 6.3, starting with the configuration suggested by **??**, which improved upon the S6/VSR2,3 by requiring exact-match coincidence, we present a procedure for

further improving the search sensitivity of the pipeline by optimizing key parameters of the search, namely the configuration of the power spectral estimation, the signal-consistency tests, the single detector SNR thresholds, and the lower frequency cutoff.

6.2 Coincident Analysis

To search for gravitational-waves from coalescing binaries, the search pipeline implements the coincident matched-filtering algorithm proposed in [**?**] A bank of post-Newtonian TaylorF2 3.5 PN order templates, generated using the metric based placement algorithm proposed in [**?**], is created to span the extended binary neutron star mass range from $1-3M\odot$. Each template, h, is filtered against the gravitationl-wave strain data of a detector, s, resulting in the matched filtering signal-to-noise ratio,

$$\rho(t) = \frac{(s|h)}{\sqrt{(h|h)}}. \tag{6.1}$$

We make use of the noise-weighted inner product

$$(a|b) = 4\int_{f_{low}}^{f_{nyquist}} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)} e^{2\pi i f t} df, \tag{6.2}$$

where $\tilde{s}$ is the Fourier transform of the data, $\tilde{h}$ is the Fourier tranform of the template waveform, and $S_n(f)$ is the power spectral density (PSD).

Excursions in the SNR timeseries are recorded as single-detector triggers if they exceed a fixed SNR threshold, and are the loudest within a fixed time window of 1 second. To cope with large number of high SNR triggers caused non-Gaussian transient events, we construct a signal-based consistency test. It is a time-frequency test, constructed by splitting the template waveform in the frequency domain amoung p bins that each contribute an equal amount of power. Each bin is filtered against the data to construct the matched filter output ρ_l. The full time-frequency χ^2 is finally constructed as

$$\chi^2 = p\sum_{l=1}^{p}\left(\frac{\rho}{p} - \rho_l\right)^2. \tag{6.3}$$

True Gravitational-wave signals, along with Gaussian-noise, return a low number for this statistic, while a noise transient, which strongly weights a limited number

of bins, would return a large χ^2 value. The value of this signal-consistency test along with the matched-filter SNR is used to construct the single detector detection statistic, $newSNR$, which is defined as

$$\rho_{new} = \begin{cases} \rho & \text{for } \chi_r^2 \leq 1 \\ \rho[\frac{1}{2}(1+(\chi_r^2)]^{-\frac{1}{6}} & \text{for } \chi_r^2 > 1, \end{cases} \tag{6.4}$$

where χ_r^2 is the reduced χ^2 statistic, $\chi_r^2 = \chi^2/(2p-2)$.

This statistic has the convenient property that for relatively quiet injections within Gaussian noise, the value is close to the simple matched-filter SNR, while also down-weighting triggers caused by non-Gaussian noise.

We require that a candidate signal be present in more than one detector. In accordance with the recommendation of [**?**], we define a coincident event as a pair of single detector triggers have the same mass and spin parameters, and are within a fixed time window of each other. This window is determined by the light-travel time between detectors, which for two-detector network of LIGO detectors we are considering is $\approx 11ms$. The single detector new ρ_{new} from the Hanford and Livingston detector, ρ_{new}^H and ρ_{new}^L, respectively are combined into a single coincident statistic given by

$$\rho_{new}^c = \sqrt{(\rho_{new}^L)^2 + (\rho_{new}^H)^2} \tag{6.5}$$

6.2.1 Significance of Candidate Events

In order to claim a candidate signal as a detection of a gravitational-wave, we need to determine the probability that it could have been a chance happening. We estimate the false alarm rate by forming coincidences between single detector triggers that are outside of the standard coincident time window.

For both computational efficiency and simplicity, we choose to form background coincidences by applying a time shift to one detector. The triggers from one detector are offset by all possible non-zero integer multiples of the a fixed interval, T_s, for which there is coincident livetime. For this analysis we choose T_s to .2 s. From all of these time slides, we collect a set of coincident triggers. As this set was formed from all of the original single detector triggers, we will refer to it as the *inclusive background*, B_{inc}.

Note, that if there is a loud gravitational-wave signal its component single detector triggers will also form coincidences that will be included within the background. The inclusive set of background triggers can be expanded as

$$B_{inc} = \{N_H * N_L\} \cup \{N_H * S_L\} \cup \{S_H * N_L\}, \tag{6.6}$$

where $N_{H/L}$ are single detector noise triggers, $S_{H/L}$ are single detector triggers from gravitational-wave signals, and $\{A*B\}$ represents the set of coincidences between the single detector triggers A and B. We can define a set of background triggers that excludes coincidences, B_{exc} , by excising single detector time surrounding each of the foreground coincident triggers, with components F_H and F_L. This can be expressed as,

$$B_{exc} = B_{inc} - \{S_H * T_L\} - \{R_H * S_L\} - \{S_H * R_L\}, \tag{6.7}$$

where $R_{L/H} \in N_{L/H}$, and the time difference between any element in $R_{L/H}$ and any element of $F_{H/L}$ is greater than the blinding window T_{blind}, which in this analysis we have chosen to be 100 ms. Although the exclusive background is likely to exclude true signals, along with a set of random coincidences due to noise, a priori it cannot be determined if any given trigger belongs to the set of signal triggers or noise triggers. As such, both backgrounds are valid for different types of questions. The inclusive background admits the possibility that all triggers could be noise generated, while the exclusive background presumes that they are signal.

Given an event wih a $\rho^c_{new} = x$, we can express the false alarm rate (FAR) as

$$FAR_{inc/exc}(x) = N_{inc/exc}(x)/T_B, \tag{6.8}$$

where $N_{bkg}(x)$ is the number of coincident events in the estimated background with a $\rho^c_{new} > x$. T_F is the coincident livetime, the cumulative amount of time in which both detectors of the two-detector network is active. T_B refers to the effective background time, which can be accurately estimated from the single-detector livetimes T_H and T_L for the Hanford and Livingston detectors, respectively, as

$$T_B = T_H * T_H/T_s \tag{6.9}$$

Note, that this is not an exact calculation of the background livetime, which can be obtained by explicitly calculating the amount of overlapping time between the two ifos for each time slide and taking the sum. Notice, that the estimate is equivelant to the exact calculation, when the start and end of chunck of analyzed data lies on multiple of the timslide interval. As such, we can calculate the upper bound on the difference between the true and estimated value of the background livetime as

$$ERROR < 2 * N_{chunks} * T_s, \tag{6.10}$$

where N_{chunks} is the number of non-contigious analysis chunks.

As our analysis discards chunks of data that are less than 2048 seconds in length, and $T_s = 200$ ms, the relative error is strictly less than .02%, and so can safely be considered negligible.

6.3 Optimizing Search Sensitivity

In this section, we retune several parameters of a CBC search, with the aim of creating a sensitive search for Binary Neutron star sources. The potential parameter space of tuning choices is quite large, so we have started with the settings that mimic the lowmass CBC search performed in S6/VSR23.

In this section, we describe the procedure for evaluating the search perfornance of a particular set of tuning parameters. The metric we will use for evaluation is sensitvity volume of the search integrated over the coincident livetime, VT, which can be expressed as,

$$VT(F) = \int \epsilon(F; r, \Omega, \Lambda, t) p(r, \Omega, \Lambda) r^2 dr d\Omega dt \tag{6.11}$$

The quanitity VT is directly proportional to the expected number of detected gravitaional-wave signals.

In the subsequent sections, we will use this metric to evaluate potential options for tuning the PSD estimation, and chisq binning.

To assess the performance of a given set of tunings, we evalute VT for 3 one-week

time spans of S6 data.

6.3.1 Power Spectrum Estimation

Because the overall sensitivity of a detector along with the shape of its power spectral density (PSD) changes over time, the spectral density used to calculated the SNR of candidate events is periodically recalculated. The S6/VSR2,3 analysis recomputed the PSD using every 1920 seconds. However, due to the additional padding required for filtering, 2048s of data was used for each PSD estimate. Each 2048s chunk of data is subdivided into 15 segments, each with 256s duration and overlapped by 50%. The PSD each chunk is calculated by first taking the median average of the Fourier transform of each segment. Finally, we truncate the inverse of the PSD in the time domain to restrict the filter corruption to a fixed length of time. In addition, this has the effect of smoothing out lines within the spectrum. An inverse truncation value of 16 seconds was used throughout S6/VSR2,3.

We investigate a straightforward improvement to this algorithm. Instead of calculating the PSD using 256 second segments and applying a 16 second inverse spectrum truncation, we propose calculating the PSD using 16 second segments directly, interpolating for the intended use case, and finally applying the same 16s inverse spectrum truncation. The results of this investigation are shown in figure Fig. 43, where the sensitive volume-time is compared for the initial reference configuration and for the proposed configuration as a function of the inverse false alarm rate. The proposed PSD estimation shows clear improvement over the initial method, resulting in an average $\approx 18\%$ increase in sensitivity between inverse false alarm rates of 10^3 and 10^4 years.

6.3.2 Signal-to-noise Threshold

For each detector, triggers are recorded when the signal-to-noise ratio exceeds a predetermined threshold, ρ_t. For the S6/VSR2,3 CBC search, only triggers with an SNR above 5.5 were recorded. Beginning with the PSD tunings proposed in sec. 6.3.1, we investigate the effect of lowering the SNR threshold to 5.0. A comparison of the search sensitivity at $\rho_t = 5.5$ and $\rho_t = 5.0$ is shown in Fig. 44. We see that lowering ρ_t from 5.5 to 5.0 has not resulted in a significant improvement in sensitivity. We

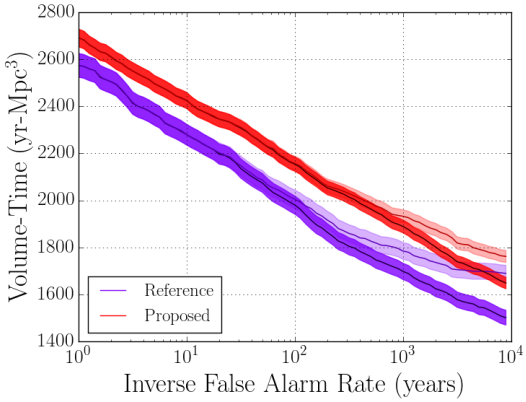

Figure 43: The combined VT as a function of inverse false alarm rate, for the combined three weeks of analysis, and for an injection population that uniformly covers the parameter space of the non-spinning BNS region, with component masses between $1 - 3M_{\odot}$. Darker colored lines indicate the inclusive IFAR value, while lighter lines show the exclusive IFAR. The reference (red) PSD estimation uses 15, 256s segments. The proposed (purple) tuning which uses 252, 16s segments. Both truncate the inverse spectrum in the time domain to 16 seconds. The proposed configuration improves the search sensitivity by $\approx 18\%$ at a false alarm rate of 1 per 1000 years.

observe that at high inverse false alarm rate, the inclusive IFAR is identical between the two thresholds, but that there is a very minor increase in sensitivity when using the exclusive IFAR.

In fig. 45 we explore where the differences between the inclusive and exclusive IFAR estimates are the greatest. At a fixed exclusive IFAR, which is directly proportional to the combined NewSNR of an injection trigger, we find that there is an inverse relation between the inclusive IFAR and the minimum single detector SNR. This indicates that lowering the SNR threshold below ≈ 5.3 will not yield an improvement in sensitivity at inclusieve false alarm rate of 1 in 1000 years, for a two-detector search composed of the Hanford and Livingston LIGO observatories. Note that this result cannot be generalized to a multi-detector network, where there can be a non-trivial increase in detection confidence due to the presence of quiet trigger in the additional detectors. Further work is required to characterize the appropriate SNR thresholds for multi-detector networks.

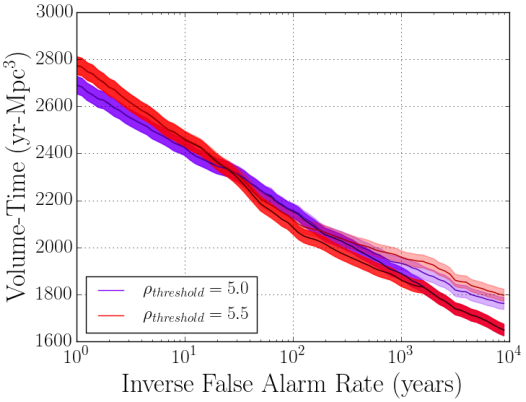

Figure 44: The combined VT as a function of inverse false alarm rate, for the combined three weeks of analysis, and for an injection population that uniformly covers the parameter space of the non-spinning BNS region, with component masses between $1 - 3M_{\odot}$. Darker colored lines indicate the inclusive IFAR value, while lighter lines show the exclusive IFAR.

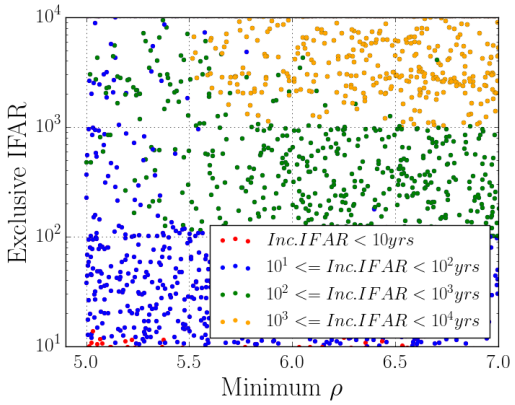

Figure 45: The distribution of Exclusive IFAR as a function of the minumum single detector SNR, for an injection population that uniformly covers the parameter space of the non-spinning BNS region, with component masses between $1 - 3M_{\odot}$. Injections are colored by the value of their inclusive IFAR. We observe that there is an inverse relationship between the inclusive IFAR and the minimum SNR value. This indicates that for a given value of inclusive IFAR, there is a corresponding SNR threshold, below which, the search sensitivity as function of inclusive IFAR will not improve

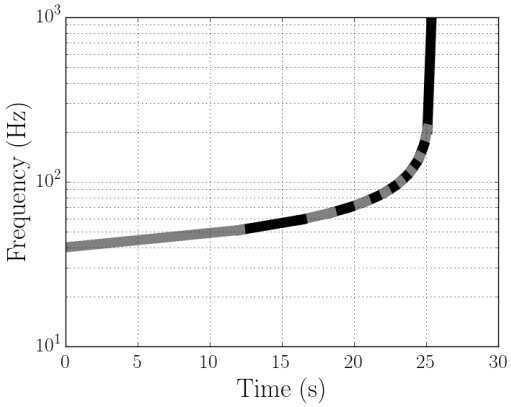

Figure 46: Approximate boundaries of the 16 bins that make up the time-frequency signal consistency test, as used in S6/VSR2,3, overlaid on the track of a $1.4 - 1.4 M_o dot$ BNS waveform.

6.3.3 Signal-consistency Test and Ranking Statistic

As detailed in sec. 6.2, the single-detector ranking statistic, ρ_{new}, is the SNR weighted by the time-frequency consistency test. The time-frequency signal consistancy test breaks a template into p bins of each power. Although the boundaries of the bins are defined in the frequency domain, as the template is a monotonic function of time and frequency, we can outline the rough time-frequency boundaries of each bin as demonstrated in fig 46, for the 16 bins used in the S6/VSR2,3 analysis. Since the response of the time-frequency chisq is dependent on the morphology of the non-guassian noise present in the data, we investigate if increasing the number of time-frequency bins for a BNS focused search, where the average template duration is significantly longer than for searches that include higher mass templates, has an effect on the search sensitivity.

Starting with the analysis tunings suggested in 6.3.2, we compare the search sensitivity at a fixed exclusive FAR of 1/1000 years. The results in Fig. 47, show that increasing the number of time-frequency bins from 16 to 64-256 there is an $\approx 12\%$ improvement in search sensitivity. From the results of Fig .48 we see that this improvement occurs at all values of the FAR.

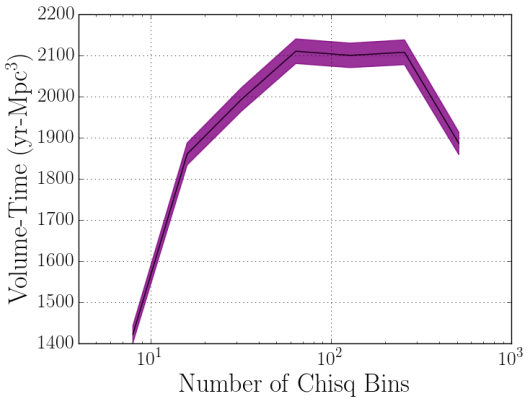

Figure 47: The combined VT at inclusive inverse false alarm rate of 1/1000 years as a function of the number of time-frequency bins in the signal-consistency test, for the combined three weeks of analysis, and for an injection population that uniformly covers the parameter space of the non-spinning BNS region, with component masses between $1 - 3M_{\odot}$. There is an $\approx 13\%$ improvement in the analysis sensitivity when using 64-256 bins, as compared to the reference 16 bins.

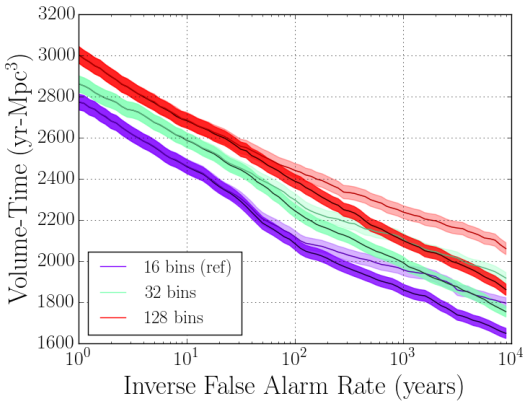

Figure 48: The combined VT as a function of inverse false alarm rate, for the combined three weeks of analysis, and for an injection population that uniformly covers the parameter space of the non-spinning BNS region, with component masses between $1 - 3M_{\odot}$. Darker colored lines indicate the inclusive IFAR value, while lighter lines show the exclusive IFAR.

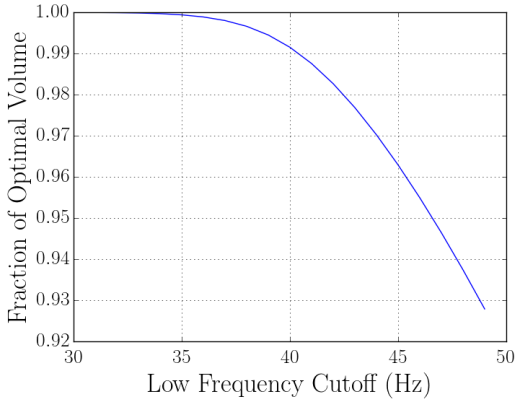

Figure 49: The fraction of the optimal search volume for a $1.4 - 1.4 M_{\odot}$ TaylorF2 BNS waveform, as a function of the lower-frequency cutoff of the matched filter.

6.3.4 Lower-frequency cutoff of the matched filter

As we have been using S6 LIGO data in the previous sectons, and expect that seismic noise will dominate at low frequencies, we have used the same 40Hz lower-frequency cutoff used in the S6/VSR2,3 analysis. We can verify that using a 40Hz lower-frequency cutoff does not impact search peformance by constructing

$$V(f_{low}) = \left[\frac{\int_{f_{low}} \frac{h^*(f)h(f)}{S_n(f)} df}{\int_0 \frac{h^*(f)h(f)}{S_n(f)} df} \right]^3 \tag{6.12}$$

where $h(f)$ is a template waveform, $S_n(f)$ is the power spectral densitity, and the quantity $V(f_{low})$ represents the fraction of the optimal volume for a single template filtered from the lower-frequency cutoff, f_{low}. Fig 49 shows that filtering from 40Hz only results in only a 1% loss in search volume, for a single 1.4-1.4$M_{\odot}$ TaylorF2 template.

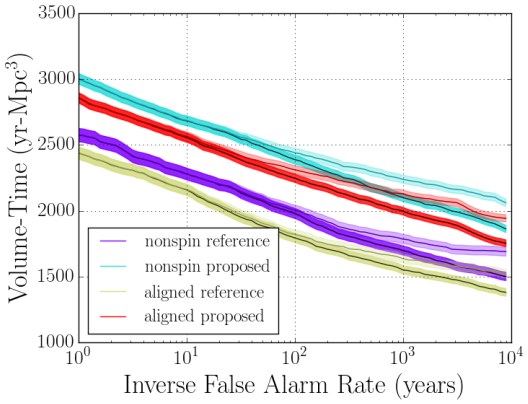

Figure 50: The combined volume - time product as a function of inverse false alarm rate, for the three sample analysis weeks, for an injection population that uniformly covers parameter space of the non-spinning BNS region, $1 - 3M_{\odot}$.

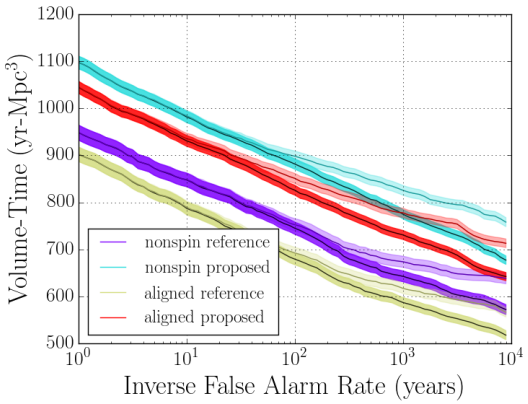

Figure 51: The combined volume - time product as a function of inverse false alarm rate, for the three sample analysis weeks, for an injection population that uniformly covers parameter space of the non-spinning BNS region, $1 - 3M_{\odot}$.

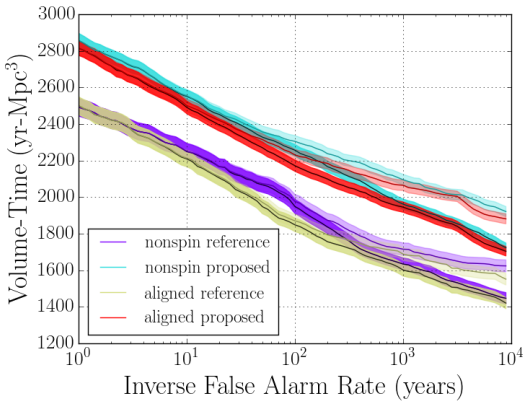

Figure 52: The combined volume - time product as a function of inverse false alarm rate, for the three sample analysis weeks, for an injection population that uniformly covers parameter space of the non-spinning BNS region, $1 - 3M_{\odot}$.

6.3.5 False Alarm Rate vs. Parameter space coverage

6.4 Sensitivity to Astrophysical Sources

6.4.1 Nonpinning injections

6.4.2 Conservative Source Distribution

6.4.3 Precessing Source

6.4.4 Alinged Spin Sources

Precession?

6.5 Conclusions

6.6 Acknowledgments

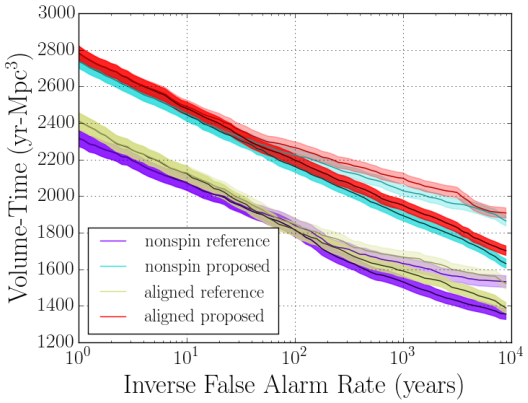

Figure 53: The combined volume - time product as a function of inverse false alarm rate, for the three sample analysis weeks, for an injection population that uniformly covers parameter space of the non-spinning BNS region, $1 - 3M_{\odot}$.