Implementing a semicoherent search for continuous gravitational waves using optimally constructed template banks

K. Wette,[1,2,*] S. Walsh,[3,2] R. Prix,[2] and M. A. Papa[2,3]

[1] *ARC Centre of Excellence for Gravitational Wave Discovery (OzGrav)*
and Centre for Gravitational Physics, Research School of Physics and Engineering,
The Australian National University, Acton ACT 2601, Australia
[2] *Max Planck Institute for Gravitational Physics (Albert Einstein Institute), D-30167 Hannover, Germany*
[3] *Department of Physics, University of Wisconsin, Milwaukee, Wisconsin 53201, USA*

(Received 10 April 2018; published 28 June 2018)

All-sky surveys for isolated continuous gravitational waves present a significant data-analysis challenge. Semicoherent search methods are commonly used to efficiently perform the computationally-intensive task of searching for these weak signals in the noisy data of gravitational-wave detectors such as LIGO and Virgo. We present a new implementation of a semicoherent search method, WEAVE, that for the first time makes full use of a parameter-space metric to generate banks of search templates at the correct resolution, combined with optimal lattices to minimize the required number of templates and hence the computational cost of the search. We describe the implementation of WEAVE and associated design choices and characterize its behavior using semianalytic models.

DOI: 10.1103/PhysRevD.97.123016

I. INTRODUCTION

The detections of short-duration gravitational-wave events from the inspiral and merger of binary black holes [1–5] and binary neutron stars [6] are enabling advances across astronomy, astrophysics, and cosmology. As the gravitational-wave detectors LIGO [7,8] and Virgo [9] improve in sensitivity in the coming years, and as new detectors KAGRA [10] and LIGO India [11] come online, it may become possible to detect gravitational radiation from other astrophysical phenomena. Rapidly-spinning, nonaxisymmetrically-deformed neutron stars will emit gravitational waves in the form of continuous quasisinusoidal signals and remain an intriguing prospect for detection with advanced instruments. Searches for continuous gravitational waves in contemporary LIGO and Virgo data are ongoing, e.g., Refs. [12–16].

Since the maximum nonaxisymmetric deformation of neutron stars is expected to be small, e.g., Ref. [17], continuous waves are expected to be weak relative to the sensitivity of the current generation of interferometric detectors. Consequentially, a significant body of research devoted to the data-analysis challenge of extracting such weak signals from the gravitational-wave detector data has accumulated. Early results [18,19] focused on the method of matched filtering the entire data set against the known continuous-wave signal model; while theoretically optimal (in the Neyman-Pearson sense), this method quickly becomes computationally intractable if some or all of the model parameters are unknown. Such is the case if one wished to target an interesting sky direction e.g., associated with a supernova remnant, e.g., Ref. [20], or a low-mass x-ray binary, e.g., Ref. [13,15], or perform an all-sky survey for isolated continuous-wave sources unassociated with known pulsars, e.g., Ref. [18]. It is the latter type of search that is the subject of this paper.

The additional challenge of a practical upper limit on the computational cost of all-sky searches has spurred the development of various suboptimal but computationally-tractable *hierarchical* or *semicoherent* algorithms [21]. They share a common approach: the data set (which for this example we assume is contiguous) with time span $\widehat{T}$ is partitioned into N segments, each with time span $\tilde{T}$. A fully-coherent matched filter search is then performed individually for each segment. Most[1] methods then combine segments by incoherently summing the power from N filters, one from each segment, which together follow a consistent frequency evolution as dictated by the continuous-wave signal model. The phase evolution need not be continuous over the N filters, however; nor do the gravitational-wave amplitudes in each segment need to be consistent. This loss of complete signal self-consistency comes, however, with a computational benefit: while the computational cost of a fully-coherent matched filter search of the entire data set scales as $\widehat{T}^n = N^n \tilde{T}^n$ with n a high

[*] karl.wette@anu.edu.au

[1] A few methods instead look for significant templates which are coincident between segments [22,23].

power ~5 to 6, the cost of a semicoherent method typically scales as $N^m \tilde{T}^n$ with $m \sim 2 \ll n$ [24]. The strain sensitivities of a fully-coherent and semicoherent search typically scale as $N^{1/2}\tilde{T}^{1/2}$ and $N^{1/4w}\tilde{T}^{1/2}$, respectively, with $w \geq 1$ [24,25]; for the loss of a factor $N^{\sim 1/4}$ in sensitivity, a semicoherent method is able to gain by being able to analyze large (e.g., $\widehat{T} \gtrsim 1$ yr) data sets, whereas a fully-coherent search would be computationally restricted to a much shorter (e.g., $\widehat{T} \ll 1$ yr) subset.

An important early advance in the development of semicoherent methods was the adaption of the Hough transform [26], originally created to analyze tracks in bubble chamber photographs, to instead track the frequency evolution of a continuous gravitational-wave signal [27]. A number of variations of the Hough transform have been implemented, which map the signal track in the time-frequency plane to either its sky position at a fixed reference frequency and frequency derivative [28] or conversely to its reference frequency and frequency derivative at a fixed sky position [29,30]. The detection statistic computed, the *number count*, sums either 0 or 1 from each segment depending on whether the significance of a filter exceeds a set threshold. Some variations use short-duration ($\tilde{T} \sim 1800$ s) segments and incoherently sum power above threshold from each segment; others analyze longer segments and set a threshold on the $\mathcal{F}$*-statistic* [19] which computes the matched filter analytically maximized over the gravitational-wave amplitudes. Another modification is to weigh each segment by the antenna response function of the detector and to sum these weights instead of 0 or 1 [31,32].

Two semicoherent methods which use short-duration segments but which, unlike the Hough transform methods, sum power without thresholding are the StackSlide [33] and PowerFlux [34] methods. The StackSlide method builds a time-frequency plane, where each column represents a segment. For each choice of signal parameters, it "slides" each column up and down in frequency so that a signal with those parameters would follow a horizontal line and then "stacks" (i.e., sums) the columns horizontally to accumulate the signal power over time for each frequency bin. (Due to this intuitive representation of a semicoherent search method, the term *StackSlide* is often used to refer to semicoherent methods in general, e.g., Ref. [24].) The PowerFlux method follows a similar methodology and in addition weights the power from each segment by that segment's noise level and antenna response function, so that segments containing transient instrumental noise and/or where the response of the detector is weak are deweighted. A "loosely-coherent" adaption to PowerFlux allows the degree of phase consistency imposed at the semicoherent stage to be controlled explicitly [35,36]. A third semicoherent method [37,38] was developed based on the observance of global correlations between search parameters [39] and uses longer segments analyzed with the $\mathcal{F}$-statistic. A comprehensive comparison of many of the all-sky search methods described above is performed in Ref. [40].

Aside from developments in semicoherent search techniques, two other ideas have played an important role in the development of continuous gravitational-wave data analysis. First is the use of a *parameter-space metric* [18,41,42], which is used to determine the appropriate resolution of the bank of template signals such that the *mismatch*, or fractional loss in the signal-to-noise ratio between any signal present in the data and its nearest template, never exceeds a prescribed maximum. The metric of the $\mathcal{F}$-statistic for continuous-wave signals was first studied rigorously in Ref. [43]. An approximate form of the metric was utilized in semicoherent search methods developed by Ref. [44], and a related approximation was used in Refs. [37,38]. The latter approximation, however, led to an underestimation of the number of required templates in the sky parameter space when analyzing long data stretches; an improved approximate metric developed in Refs. [45,46] addresses this limitation. It was also later realized that a further approximation fundamental to the metric derivation—namely that the prescribed maximum mismatch (as measured by the metric) could be assumed small—generally does not hold under realistic restrictions on computational cost. This issue was addressed in Ref. [47], which computed an empirical relation between the metric-measured mismatch and the true mismatch of the $\mathcal{F}$-statistic.

A second important idea is the borrowing of results from lattice theory, e.g., Ref. [48], to optimize the geometric placement of templates within the search parameter space, so as to fulfill the maximum prescribed mismatch criteria described above with the smallest possible density of templates [49,50]. Practical algorithms for generating template banks for continuous-wave searches, using both the parameter-space metric and optimal lattices, were proposed in Refs. [51,52]. An alternative idea studied in Refs. [53,54] is to instead place templates at random, using the parameter-space metric only as a guide as to the relative density of templates; this idea has found utility in searches for radio, e.g., [55] and x-ray, e.g., Ref. [56], pulsars.

The number of computations that must be performed during an all-sky search, even when utilizing an efficient semicoherent search method, remains formidable. For example, a recent all-sky search [16] of data from the first Advanced LIGO observing run divided the data into $N = 12$ segments of time span $\tilde{T} = 210$ hr, performed $\sim 3 \times 10^{15}$ matched-filtering operations per segment, and finally performed $\sim 3 \times 10^{17}$ incoherent summations to combine filter power from each segment. The total computational cost of the search was $\sim 6 \times 10^5$ CPU days, although this was distributed over $\mathcal{O}(10^4)$ computers volunteered through the Einstein@Home distributed computing project [57]. Nevertheless, the significant number of filtering/incoherent summation operations that must be performed during a typical all-sky search emphasizes the need to optimize the construction of the template banks, and thereby minimize the computational cost of the search, as much as is practicable.

In this paper, we present WEAVE, an implementation of a semicoherent search method for continuous gravitational waves. This implementation brings together, for the first time, several strands of previous research: the use of a semicoherent method to combine data segments analyzed with the $\mathcal{F}$-statistic, combined with optimal template placement using the parameter-space metric of Refs. [45,46] and optimal lattices [52]. After a review of relevant background information in Sec. II, the WEAVE implementation is presented in Sec. III. In Sec. IV, we demonstrate that important behaviors of the WEAVE implementation can be modeled semianalytically, thereby enabling characterization and optimization of a search setup without, in the first instance, the need to resort to time-consuming Monte Carlo simulations. In Sec. V, we discuss ideas for further improvement and extension.

II. BACKGROUND

This section presents background material pertaining to the continuous-wave signal model, parameter-space metric, and template bank generation.

A. Continuous-wave signals

The phase of a continuous-wave signal $\phi(t, \boldsymbol{\lambda})$ at time t at the detector is given by, neglecting relativistic corrections [19],

$$\frac{\phi(t,\boldsymbol{\lambda})}{2\pi} \approx \sum_{s=0}^{s_{\max}} f^{(s)} \frac{(t-t_0)^{s+1}}{(s+1)!} + \frac{\boldsymbol{r}(t)\cdot\boldsymbol{n}}{c} f_{\max}. \qquad (1)$$

The first term on the right-hand side primarily[2] encodes the loss of rotational energy of the neutron star as observed from the Solar System barycenter: f^0 is the gravitational-wave frequency, and the *spin-downs* f^1, f^2, etc., are the first-order, second-order, etc., rates of change of the gravitational-wave frequency with time. All $f^{(s)}$ parameters are given with respect to a reference time t_0. The second term on the right-hand side describes the Doppler modulation of the gravitational waves due to the motion of an Earth-based detector: $\boldsymbol{r}(t)$ is the detector position relative to the Solar System barycenter, thereby including both the sidereal and orbital motions of the Earth, and $\boldsymbol{n}$ is a unit vector pointing from the Solar System barycenter to the continuous-wave source. The value of $f_{\max}$ is chosen conservatively to be the maximum of $f(t) \equiv d\phi(t,\boldsymbol{\lambda})/dt$ over the time span of the analyzed data.

Together, the *phase evolution parameters* $\boldsymbol{\lambda} = (\boldsymbol{n}, f^{(s)})$ parametrize the continuous-wave signal template; additional *amplitude parameters* $\boldsymbol{\mathcal{A}}$ are analytically maximized over when computing the $\mathcal{F}$-statistic [19]. In noise, the $\mathcal{F}$-statistic is a central χ^2-statistic with 4 degrees of freedom; when in the vicinity of a signal, the noncentrality parameter $\tilde{\rho}^2$ of the noncentral χ^2 distribution scales as $\tilde{\rho}^2 \propto h_0^2 T/S_h[f(t)]$, where h_0 is the gravitational-wave amplitude, T is the amount of analyzed data, and $S_h[f(t)]$ is the noise power spectral density in the vicinity of the signal frequency $f(t)$.

B. Parameter-space metric

The parameter-space metric $\mathbf{g}$ of the $\mathcal{F}$-statistic is defined by a second-order Taylor expansion of the noncentrality parameter,

$$\rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda}) = \rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda}^{\mathrm{s}}) \times [1 - g_{ij}(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}})\Delta\lambda_i \Delta\lambda_j] + \mathcal{O}(\Delta\lambda^3), \qquad (2)$$

with implicit summation over i, j and where

$$g_{ij}(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}) \equiv \frac{-1}{2\rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda})} \frac{\partial^2 \rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda})}{\partial\lambda_i \partial\lambda_j}\bigg|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^{\mathrm{s}}}. \qquad (3)$$

Here, $\rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda}^{\mathrm{s}})$ is the noncentrality parameter of the $\mathcal{F}$-statistic when perfectly matched to a signal with parameters $\boldsymbol{\lambda}^{\mathrm{s}}$, and $\rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda})$ is the noncentrality parameter when computed at some mismatched parameters $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathrm{s}} + \Delta\boldsymbol{\lambda}$. The mismatch is defined to be

$$\mu(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda}) \equiv 1 - \frac{\rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda})}{\rho^2(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}}; \boldsymbol{\lambda}^{\mathrm{s}})} \qquad (4)$$

$$= g_{ij}(\boldsymbol{\mathcal{A}}, \boldsymbol{\lambda}^{\mathrm{s}})\Delta\lambda_i \Delta\lambda_j. \qquad (5)$$

A very useful approximation to Eq. (3) is the *phase metric* [18,43,44]; it discards the amplitude modulation of the signal and thereby the dependence on the known parameters $\mathcal{A}$, retaining only dependence on the phase evolution parameters:

$$g_{ij}(\boldsymbol{\lambda}^{\mathrm{s}}) \equiv \left\langle \frac{\partial\phi}{\partial\lambda_i} \frac{\partial\phi}{\partial\lambda_j} \right\rangle - \left\langle \frac{\partial\phi}{\partial\lambda_i} \right\rangle \left\langle \frac{\partial\phi}{\partial\lambda_j} \right\rangle. \qquad (6)$$

C. Optimal template placement

Template placement using optimal lattices is an example of a *sphere covering*, e.g., Ref. [48]: a collection of lattice-centered n-dimensional spheres of equal radius. The radius is chosen to be the smallest value that satisfies the property that each point in the n-dimensional parameter space is contained in at least one sphere. A lattice where the ratio of the volume of the sphere to the volume of a lattice cell is

[2] The rate of spin-down observed at the Solar System barycenter is strictly a combination of the spin-down observed in the source frame and the motion of the source [19]; the latter is usually assumed to be small.

minimized generates a minimal sphere covering, i.e., the minimal number of points required to cover a parameter space, which is exactly the property desired for template banks. (For example, in two dimensions, the minimal sphere covering is generated by the hexagonal lattice.) We identify the covering spheres with the *metric ellipsoids* $g_{ij}(\boldsymbol{\lambda}^{\mathrm{s}})\Delta\lambda_i\Delta\lambda_j \le \mu_{\mathrm{max}}$, where μ_{max} is the prescribed maximum; it follows that the radius of the covering spheres is $\sqrt{\mu_{\mathrm{max}}}$. A matrix transform $\mathbf{T}$ can then be constructed [52], which takes integers in $\boldsymbol{\xi} \in \mathbb{Z}^n$ to template parameters $\boldsymbol{\lambda}$ to generate the template bank,

$$\lambda_i = T_{ij}\xi_j = B_{ik}[\mathbf{g}(\boldsymbol{\lambda}^{\mathrm{s}})]L_{kj}\xi_j, \tag{7}$$

where $\mathbf{B}$ is a function of the metric $\mathbf{g}(\boldsymbol{\lambda}^{\mathrm{s}})$ and L is particular to the lattice being used. If $\mathbf{T}$ is a lower triangular matrix, an efficient algorithm [52] can be found for generating the template bank.

D. Reduced supersky metric

In order for Eq. (7) to preserve the sphere covering property, however, it must be independent of the template parameters $\boldsymbol{\lambda}^{\mathrm{s}}$. Since $\mathbf{B}$ is a function of the metric, we require a metric which is also independent of λ: $\mathbf{g}(\boldsymbol{\lambda}^{\mathrm{s}}) \to \mathbf{g}$. The phase metric of Eq. (6) is independent of the frequency and spin-down parameters $f^{(s)}$ but retains a dependence on sky position parameters, e.g., $\mathbf{g}(\boldsymbol{\lambda}^{\mathrm{s}}) \to \mathbf{g}(\alpha,\delta)$ in terms of right ascension α and declination δ. The question of how to derive a useful metric which is independent of the sky position parameters, i.e., $\mathbf{g}(\alpha,\delta) \to \mathbf{g}$, has stimulated numerous approaches, e.g., Refs. [37,44,58]. In Ref. [45], a useful $\mathbf{g}$ is derived through the following procedure:

(i) $\mathbf{g}(\alpha,\delta)$ is expressed in terms of the three components of $\boldsymbol{n} = (n_x, n_y, n_z)$, instead of two parameters such as (α,δ). The three components of $\boldsymbol{n}$ are taken to be independent; geometrically, this is equivalent to embedding $\mathbf{g}(\alpha,\delta)$ into a three-dimensional *supersky* parameter space, instead of being restricted to the 2-sphere defined by (α,δ). In the supersky parameter space, $\mathbf{g}$ is independent of the sky position parameters; i.e., we have the desired $\mathbf{g}(\alpha,\delta) \to \mathbf{g}$, but with the addition of a third unwanted parameter-space dimension.

(ii) A linear coordinate transform $(n_x, n_y, n_z, f^{(s)}) \to (n_a, n_b, n_c, \nu^{(s)})$ is derived which satisfies the following: $\mathbf{g}$ is diagonal in the sky position parameters (n_a, n_b, n_c), i.e., $g_{n_a n_b} = g_{n_a n_c} = g_{n_b n_c} = 0$; $g_{n_a n_a} \gg g_{n_c n_c}$; and $g_{n_b n_b} \gg g_{n_c n_c}$. The last two properties imply that the metric ellipsoids are much longer along the n_c axis than along the n_a and n_b axes. In computing the coordinate transform, use is made of the well-known correlation between the sky and frequency/spin-down parameters of the continuous-wave signal, e.g., Refs. [39,59]. The correlations arise because, on sufficiently short timescales, the change in phase due to the cyclical sidereal and orbital motions of the Earth may be Taylor expanded as linear, quadratic, etc., changes in phase with time and thereby are equivalent to changes in the frequency ($f^{(0)} \equiv f$), first spin-down ($f^{(1)} \equiv \dot{f}$), etc., parameters.

(iii) Since, in the new coordinates $(n_a, n_b, n_c, \nu^{(s)})$ the mismatch μ is only weakly dependent on $g_{n_c n_c}$, a useful approximate metric $\mathbf{g}$ is found by discarding the n_c dimension. Geometrically, this corresponds to projecting the three-dimensional supersky parameter space and metric onto the two-dimensional n_a–n_b plane. The resultant *reduced supersky* parameter-space metric $\mathbf{g}$ and associated coordinates $(n_a, n_b, \nu^{(s)})$ have reduced the sky parameter space dimensionality back to 2, while retaining the property that $\mathbf{g}$ is parameter independent.

III. WEAVE IMPLEMENTATION

This section describes the WEAVE implementation of the semicoherent search method, a schematic of which is shown in Fig. 1. The implementation is freely available as part of the LALSUITE [60] gravitational-wave data-analysis library.

A. Overview

In step 1, the user runs a precursor program LALAPPS_WEAVESETUP, which takes as an argument a list of N segments $\{(t_\ell^{\mathrm{start}}, t_\ell^{\mathrm{stop}})\}_{\ell=0}^{N-1}$ into which the data set is to be partitioned. The program computes in step 2 the N coherent parameter-space metrics $\widetilde{\mathbf{g}_\ell}$ used to construct template banks within each segment and the semicoherent parameter-space metric $\widehat{\mathbf{g}}$ used to incoherently combine segments. The metrics are written to a *setup file* in the Flexible Image Transport System format [61]. Due to the numerical ill-conditioned-ness of the parameter-space metric [43,45], this computation involves a boot-strapping process, whereby successively better-conditioned iterations of the supersky metric are computed, before then computing the reduced supersky metric as outlined in Sec. II D. Since this boot-strapping process can be time consuming for large N, and may give slightly different results on different computer hardware, precomputing the metrics both saves computing time and adds robustness against numerical errors. Note that, by Eq. (1), the sky components of the metrics will scale with f_{max}^2; since its value depends on the search frequency parameter space, which is not known by LALAPPS_WEAVESETUP, an arbitrary fiducial value f_{fiducial} is used, and the sky components of the metrics are later rescaled by $(f_{\mathrm{max}}/f_{\mathrm{fiducial}})^2$.

In step 3, the user runs the main search program LALAPPS_WEAVE. The principle arguments to this program are the setup file output by LALAPPS_WEAVESETUP, the search parameter space, and the prescribed maximum mismatches

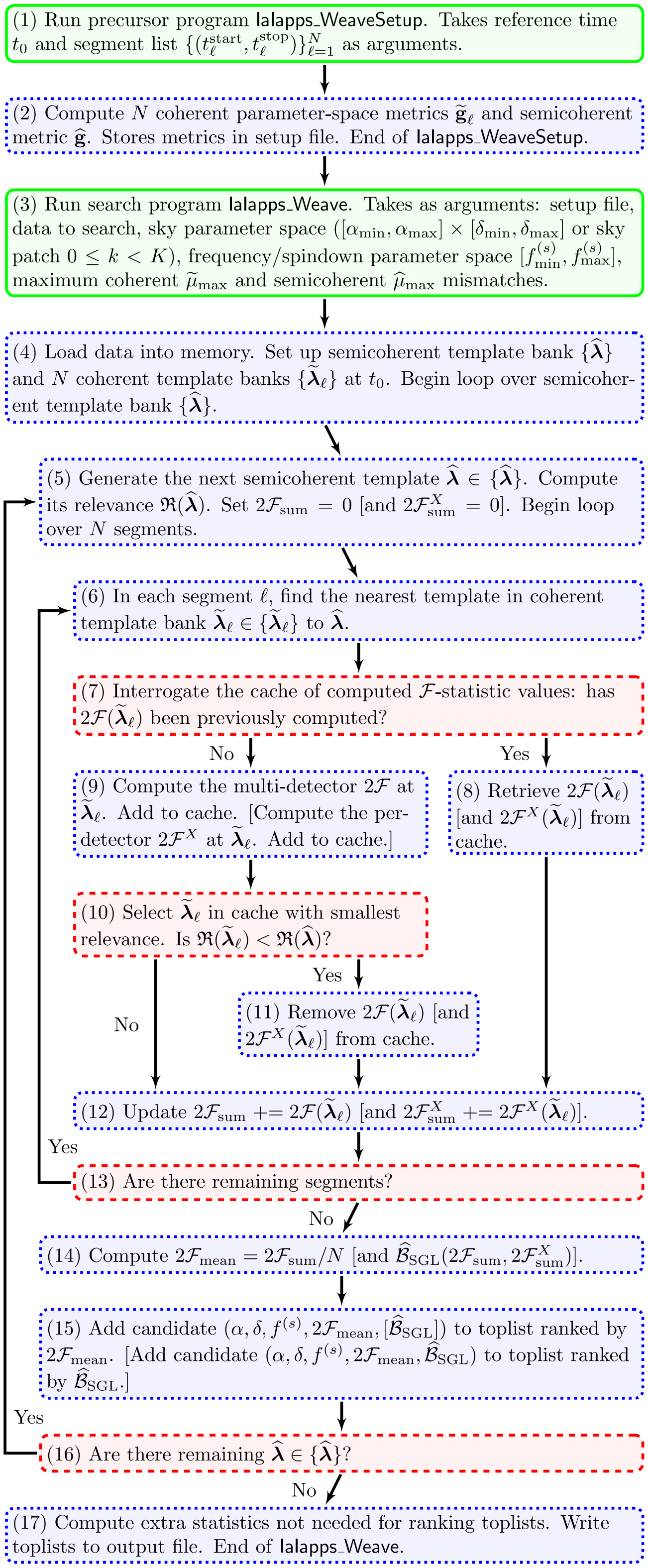

(1) Run precursor program lalapps_WeaveSetup. Takes reference time t_0 and segment list $\{(t_\ell^{\mathrm{start}}, t_\ell^{\mathrm{stop}})\}_{\ell=1}^N$ as arguments.
(2) Compute N coherent parameter-space metrics $\tilde{\mathbf{g}}_\ell$ and semicoherent metric $\hat{\mathbf{g}}$. Stores metrics in setup file. End of lalapps_WeaveSetup.
(3) Run search program lalapps_Weave. Takes as arguments: setup file, data to search, sky parameter space ($[\alpha_{\min}, \alpha_{\max}] \times [\delta_{\min}, \delta_{\max}]$ or sky patch $0 \le k < K$), frequency/spindown parameter space $[f^{(s)}_{\min}, f^{(s)}_{\max}]$, maximum coherent $\tilde{\mu}_{\max}$ and semicoherent $\hat{\mu}_{\max}$ mismatches.
(4) Load data into memory. Set up semicoherent template bank $\{\hat{\boldsymbol{\lambda}}\}$ and N coherent template banks $\{\tilde{\boldsymbol{\lambda}}_\ell\}$ at t_0. Begin loop over semicoherent template bank $\{\hat{\boldsymbol{\lambda}}\}$.
(5) Generate the next semicoherent template $\hat{\boldsymbol{\lambda}} \in \{\hat{\boldsymbol{\lambda}}\}$. Compute its relevance $\mathfrak{R}(\hat{\boldsymbol{\lambda}})$. Set $2\mathcal{F}_{\mathrm{sum}} = 0$ [and $2\mathcal{F}^X_{\mathrm{sum}} = 0$]. Begin loop over N segments.
(6) In each segment ℓ, find the nearest template in coherent template bank $\tilde{\boldsymbol{\lambda}}_\ell \in \{\tilde{\boldsymbol{\lambda}}_\ell\}$ to $\hat{\boldsymbol{\lambda}}$.
(7) Interrogate the cache of computed $\mathcal{F}$-statistic values: has $2\mathcal{F}(\tilde{\boldsymbol{\lambda}}_\ell)$ been previously computed?
No
Yes
(9) Compute the multi-detector $2\mathcal{F}$ at $\tilde{\boldsymbol{\lambda}}_\ell$. Add to cache. [Compute the per-detector $2\mathcal{F}^X$ at $\tilde{\boldsymbol{\lambda}}_\ell$. Add to cache.]
(8) Retrieve $2\mathcal{F}(\tilde{\boldsymbol{\lambda}}_\ell)$ [and $2\mathcal{F}^X(\tilde{\boldsymbol{\lambda}}_\ell)$] from cache.
(10) Select $\tilde{\boldsymbol{\lambda}}_\ell$ in cache with smallest relevance. Is $\mathfrak{R}(\tilde{\boldsymbol{\lambda}}_\ell) < \mathfrak{R}(\hat{\boldsymbol{\lambda}})$?
Yes
No
(11) Remove $2\mathcal{F}(\tilde{\boldsymbol{\lambda}}_\ell)$ [and $2\mathcal{F}^X(\tilde{\boldsymbol{\lambda}}_\ell)$] from cache.
(12) Update $2\mathcal{F}_{\mathrm{sum}} \mathrel{+}= 2\mathcal{F}(\tilde{\boldsymbol{\lambda}}_\ell)$ [and $2\mathcal{F}^X_{\mathrm{sum}} \mathrel{+}= 2\mathcal{F}^X(\tilde{\boldsymbol{\lambda}}_\ell)$].
Yes
(13) Are there remaining segments?
No
(14) Compute $2\mathcal{F}_{\mathrm{mean}} = 2\mathcal{F}_{\mathrm{sum}}/N$ [and $\hat{\mathcal{B}}_{\mathrm{SGL}}(2\mathcal{F}_{\mathrm{sum}}, 2\mathcal{F}^X_{\mathrm{sum}})$].
(15) Add candidate $(\alpha, \delta, f^{(s)}, 2\mathcal{F}_{\mathrm{mean}}, [\hat{\mathcal{B}}_{\mathrm{SGL}}])$ to toplist ranked by $2\mathcal{F}_{\mathrm{mean}}$. [Add candidate $(\alpha, \delta, f^{(s)}, 2\mathcal{F}_{\mathrm{mean}}, \hat{\mathcal{B}}_{\mathrm{SGL}})$ to toplist ranked by $\hat{\mathcal{B}}_{\mathrm{SGL}}$.]
Yes
(16) Are there remaining $\hat{\boldsymbol{\lambda}} \in \{\hat{\boldsymbol{\lambda}}\}$?
No
(17) Compute extra statistics not needed for ranking toplists. Write toplists to output file. End of lalapps_Weave.

FIG. 1. Schematic of the WEAVE implementation. Boxes with solid borders (green) represent actions taken by the user. Boxes with dotted borders (blue) represent actions taken by the program. Boxes with dashes borders (red) represent decisions the program must take. Bracketed text denotes the optional computation of $\hat{\mathcal{B}}_{\mathrm{SGL}}$. See the text in Sec. III for a full description of the WEAVE implementation.

$\tilde{\mu}_{\max}$ and $\hat{\mu}_{\max}$ for the coherent and semicoherent template banks, respectively. The frequency and spin-down parameter space are specified by ranges $[f^{(s)}_{\min}, f^{(s)}_{\max}]$, where $s = 0$, 1, etc., as required. The sky search parameter space may be specified either as a rectangular patch in right ascension and declination $[\alpha_{\min}, \alpha_{\max}] \otimes [\delta_{\min}, \delta_{\max}]$ or alternatively partitioned into K patches containing an approximately equal number of templates (see the Appendix) and a patch selected by an index k, $0 \le k < K$. In step 4, various preparatory tasks are performed, such as loading the gravitational-wave detector data into memory, before beginning the main search loop.

The main search loop of a semicoherent search method may be structured in two complementary ways, which differ in the memory each requires to store intermediate results:

(i) The semicoherent template bank $\{\hat{\boldsymbol{\lambda}}\}$ is stored in memory, and the N segments are processed in sequence. For each segment ℓ, every coherent template $\tilde{\boldsymbol{\lambda}}_\ell \in \{\tilde{\boldsymbol{\lambda}}_\ell\}$ is mapped back to the semicoherent template bank, i.e., $\tilde{\boldsymbol{\lambda}}_\ell \to \hat{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\lambda}}_\ell)$. Because the semicoherent template bank must track the continuous-wave signal over a larger time span $\hat{T} \gg \tilde{T}$ than the coherent template banks, it will contain a greater density of templates; the ratio of semicoherent to coherent template bank densities is the *refinement factor* $\gamma \ge 1$ [38,46]. It follows that the mapping $\tilde{\boldsymbol{\lambda}}_\ell \to \hat{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\lambda}}_\ell)$ will be one to many.

As the N segments are processed, any semicoherent detection statistic associated with $\hat{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\lambda}}_\ell)$ is then updated based on the corresponding coherent detection statistic associated with $\tilde{\boldsymbol{\lambda}}_\ell$. For example, it is common to compute the summed $\mathcal{F}$-statistic $2\mathcal{F}_{\mathrm{sum}} \equiv \sum_{\ell=0}^{N-1} 2\mathcal{F}(\tilde{\boldsymbol{\lambda}}_\ell)$; here, we would then have $2\mathcal{F}_{\mathrm{sum}}[\hat{\boldsymbol{\lambda}}(\tilde{\boldsymbol{\lambda}}_\ell)] + = 2\mathcal{F}(\tilde{\boldsymbol{\lambda}}_\ell)$. Once every segment has been processed, computed $2\mathcal{F}_{\mathrm{sum}}$ for every $\hat{\boldsymbol{\lambda}} \in \{\hat{\boldsymbol{\lambda}}\}$ will exist in memory. The memory usage of the main search loop will therefore be proportional to the number of semicoherent templates $\hat{\mathcal{N}} \equiv \gamma \times \langle \tilde{\mathcal{N}} \rangle$, where $\langle \tilde{\mathcal{N}} \rangle$ is the average number of templates in a coherent template bank.

(ii) The N coherent template banks $\{\tilde{\boldsymbol{\lambda}}_\ell\}$ are stored in memory, and the semicoherent template bank is processed in sequence. Each semicoherent template $\hat{\boldsymbol{\lambda}} \in \{\hat{\boldsymbol{\lambda}}\}$ is mapped back to the coherent template bank in each segment ℓ, i.e., $\hat{\boldsymbol{\lambda}} \to \tilde{\boldsymbol{\lambda}}_\ell(\hat{\boldsymbol{\lambda}})$; since $\hat{\mathcal{N}} \ge \tilde{\mathcal{N}}$ in each segment, this mapping will be many to one. With these N mappings in hand, the semicoherent detection statistics may be immediately computed in full, e.g., $2\mathcal{F}_{\mathrm{sum}}(\hat{\boldsymbol{\lambda}}) = \sum_{\ell=0}^{N-1} 2\mathcal{F}[\tilde{\boldsymbol{\lambda}}_\ell(\hat{\boldsymbol{\lambda}})]$. The memory usage of the main search loop will therefore be proportional to $N \times \langle \tilde{\mathcal{N}} \rangle$.

For the parameter-space metric for all-sky searches, $\gamma \gg N$ [38,46], and therefore the latter structuring given above will have the lower memory requirement; the WEAVE implementation uses this structuring of the main search loop. The semicoherent template bank $\{\hat{\boldsymbol{\lambda}}\}$ is generated one template at a time using the algorithm described in Ref. [52]. For each coherent template bank, an efficient lookup table [52] is constructed for the mapping $\hat{\boldsymbol{\lambda}} \rightarrow \tilde{\boldsymbol{\lambda}}_{\ell}(\hat{\boldsymbol{\lambda}})$.

We note an important distinction between the definition of the WEAVE template banks and the traditional StackSlide picture of a semicoherent search method. In the latter picture, the frequency and spin-down template banks of each segment are defined with respect to *individual* reference times $(t_0)_{\ell}$, typically the midtime of each segment. When combining segments, therefore, the frequency and spin-down parameters of each coherent template must be adjusted so as to bring the parameters of all segments to a common reference time t_0; this is the "sliding" step. The WEAVE implementation, however, defines the frequency and spin-down templates banks of all segments at the *same* reference time t_0, which is also the reference time of the semicoherent bank. Consequentially, there is no analogy to the sliding step of StackSlide. Instead, the orientation of the metric ellipses in the $(f, \dot{f})$ plane changes from segment to segment, as illustrated in Fig. 2. As the absolute difference

$$|\Delta_{\ell}| = |t_{\ell}^{\mathrm{mid}} - t_0| \tag{8}$$

between the midtime of each segment $t_{\ell}^{\mathrm{mid}} \equiv (t_{\ell}^{\mathrm{start}} + t_{\ell}^{\mathrm{stop}})/2$ and t_0 increases, both the extent of the ellipses in f and the correlation between f and $\dot{f}$ also increase.

Steps 5–16 comprise the main search loop, which performs two key tasks: the computation and output of the detection statistics over the semicoherent template bank (steps 5, 6, and 12–17) and the management of an internal cache of required detection statistics computed on each coherent template bank (steps 7–11). These two tasks are described more fully in the following two sections and with reference to a diagram of their operation in Fig. 3.

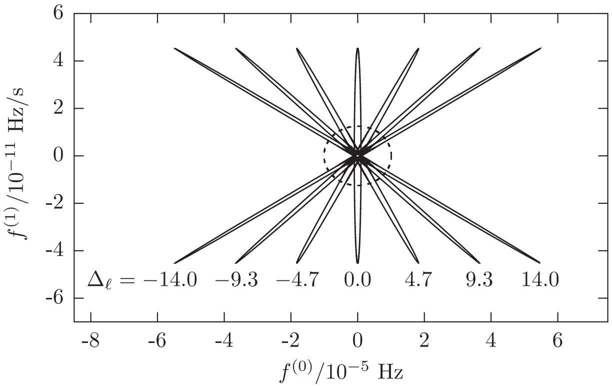


FIG. 2. Coherent (solid) and semicoherent (dashed) metric ellipses in the $(f, \dot{f})$ plane. The seven coherent metric ellipses are for segments with $\tilde{T} = 2$ days, evenly spaced within a time span of $\hat{T} = 30$ days; each ellipse is labeled by its value of $\Delta_{\ell}/\mathrm{days}$ [see Eq. (8)]. The reference time t_0 for all metrics is centered within $\hat{T}$. The coherent and semicoherent metric ellipses are plotted at maximum mismatches of $\tilde{\mu}_{\max} = 0.1$ and $\hat{\mu}_{\max} = 10$, respectively.

In this section and in Fig. 1, we focus for simplicity on the computation of the semicoherent $\mathcal{F}$-statistics $2\mathcal{F}_{\mathrm{sum}}$ and $2\mathcal{F}_{\mathrm{mean}} \equiv 2\mathcal{F}_{\mathrm{sum}}/N$. The computation of other detection statistics is also possible; in particular, a family of Bayesian statistics has been developed which weigh the likelihood of a continuous-wave signal against that of an instrumental line which appears in all segments [62,63] or a transient instrumental line which appears only in one segment [64]. Computation of the former statistic, denoted $\hat{\mathcal{B}}_{\mathrm{SGL}}$, is also illustrated in Fig. 1; it takes as input the multidetector $2\mathcal{F}_{\mathrm{sum}}$ which uses data from all gravitational-wave detectors, as well as the per-detector $2\mathcal{F}_{\mathrm{sum}}^{X}$ which are computed from each detector X individually.

B. Computation of semicoherent statistics

In steps 5 and 16 (Fig. 1), the main loop of the search method generates successive points $\hat{\boldsymbol{\lambda}}$ in the semicoherent template bank. An example of such a point is indicated in Fig. 3. Next, in steps 6 and 13, each segment ℓ is visited, and the mapping $\hat{\boldsymbol{\lambda}} \rightarrow \tilde{\boldsymbol{\lambda}}_{\ell}(\hat{\boldsymbol{\lambda}})$ is performed. The mapping used by WEAVE is nearest-neighbor interpolation: the $\hat{\boldsymbol{\lambda}}$ is expressed in the coherent metric coordinates of the ℓth segment, and the nearest (with respect to the metric) coherent template in the respective bank $\tilde{\boldsymbol{\lambda}}_{\ell}(\hat{\boldsymbol{\lambda}})$ is determined. If the template bank is constructed on a lattice, efficient algorithms exist for determining the nearest point, e.g., Ref. [52] and references therein. In Fig. 3, example nearest coherent templates are labeled $\tilde{\boldsymbol{\lambda}}_1(\hat{\boldsymbol{\lambda}})$, $\tilde{\boldsymbol{\lambda}}_2(\hat{\boldsymbol{\lambda}})$, and $\tilde{\boldsymbol{\lambda}}_3(\hat{\boldsymbol{\lambda}})$.

As each nearest point is determined, the coherent $\mathcal{F}$-statistic in the respective segment is computed (steps 7–11; see Sec. III C), and the value of the semicoherent statistic $2\mathcal{F}_{\mathrm{sum}}$ is updated (step 12). Once all segments have been processed (step 13), additional semicoherent statistics such as $2\mathcal{F}_{\mathrm{mean}}$ are computed (step 14), and a *candidate* comprising the signal parameters together with the computed semicoherent statistics is added (step 15) to one or more *top lists* which rank[3] each candidate by a chosen semicoherent statistic. The size of the top lists is generally of a fixed user-determined size so that only a fixed number of the most promising candidates will be returned.

Once the semicoherent template bank is exhausted (step 16), the top lists are written to an output file in the Flexible Image Transport System format, and the search concludes (step 17).

[3] Top lists are implemented efficiently as a binary heap, e.g., Ref. [65].

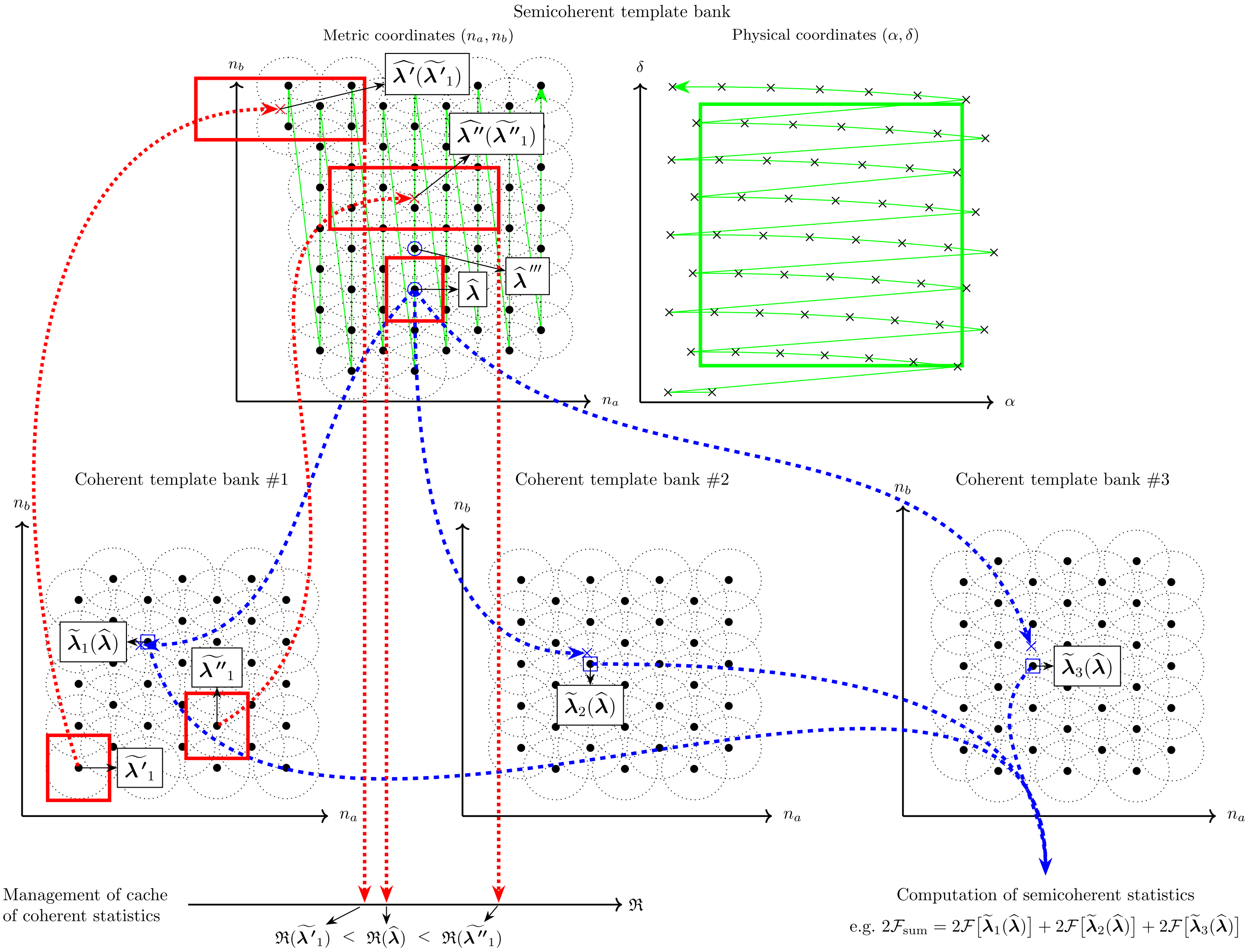

Semicoherent template bank
Metric coordinates (n_a, n_b)
Physical coordinates (α, δ)
n_b
n_a
δ
α
$\widehat{\lambda}'(\widetilde{\lambda}'_1)$
$\widehat{\lambda}''(\widetilde{\lambda}''_1)$
$\widehat{\lambda}'''$
$\widehat{\lambda}$
Coherent template bank #1
Coherent template bank #2
Coherent template bank #3
$\widetilde{\lambda}_1(\widehat{\lambda})$
$\widetilde{\lambda}''_1$
$\widetilde{\lambda}'_1$
$\widetilde{\lambda}_2(\widehat{\lambda})$
$\widetilde{\lambda}_3(\widehat{\lambda})$
Management of cache of coherent statistics
$\mathfrak{R}$
$\mathfrak{R}(\widetilde{\lambda}'_1) < \mathfrak{R}(\widehat{\lambda}) < \mathfrak{R}(\widetilde{\lambda}''_1)$
Computation of semicoherent statistics
e.g. $2\mathcal{F}_{\text{sum}} = 2\mathcal{F}[\widetilde{\lambda}_1(\widehat{\lambda})] + 2\mathcal{F}[\widetilde{\lambda}_2(\widehat{\lambda})] + 2\mathcal{F}[\widetilde{\lambda}_3(\widehat{\lambda})]$

FIG. 3. Diagram of the WEAVE template banks for an example search setup with three segments. Shown are the semicoherent and coherent template banks in the two sky parameters of the reduced supersky metric (n_a, n_b); the frequency and spin-down dimensions are omitted. For clarity, the semicoherent and coherent template banks are plotted with metric ellipses of the same size. Arrows with solid lines (green) represent iteration over the semicoherent template bank. Arrows with dashed lines (blue) represent the process of computing semicoherent statistics, described in Sec. III B. Arrows with dotted lines (red) represent the process of managing the caches of coherent statistics, described in Sec. III C. Also shown is the semicoherent template bank in physical coordinates (α, δ); the solid square (green) shows the rectangular boundary of the sky parameter space in these coordinates.

C. Management of cache of coherent statistics

It is important that the main search loop minimizes its memory usage as much as possible. Even though in Sec. III A we chose a structuring of the main search loop so as to reduce memory usage, a naive implementation which stores $N \times \langle\widetilde{\mathcal{N}}\rangle$ coherent statistics would still require a prohibitive amount of memory, given that both N and $\langle\widetilde{\mathcal{N}}\rangle$ are typically large. We therefore implement a per-segment cache which stores only those coherent statistics associated with coherent templates $\widetilde{\lambda}_\ell$ accessible from the unprocessed portion of the semicoherent template bank via the mapping $\widehat{\lambda} \to \widetilde{\lambda}_\ell(\widehat{\lambda})$. Put another way, if a $\widetilde{\lambda}_\ell$ can no longer be mapped to by any $\widehat{\lambda}$ remaining in $\{\widehat{\lambda}\}$, then $2\mathcal{F}(\widetilde{\lambda}_\ell)$ can be safely removed from the cache.

In order to devise a cache management algorithm with the above desired properties, we first define an operator called *relevance*, denoted $\mathfrak{R}: \lambda \to \mathbb{R}$. The relevance operates on both coherent and semicoherent templates and should satisfy the following property:

$$\begin{aligned}&\text{For all } \widetilde{\lambda}_\ell \in \{\widetilde{\lambda}_\ell\} \text{ and for all } \hat{\lambda} \in \{\hat{\lambda}\}, \text{ the condition}\\ &\mathfrak{R}(\widetilde{\lambda}_\ell) < \mathfrak{R}(\hat{\lambda}) \text{ implies that no mapping}\\ &\hat{\lambda} \to \widetilde{\lambda}_\ell(\hat{\lambda}) \text{ exists in the remaining } \{\hat{\lambda}\}, \text{ and thus}\\ &2\mathcal{F}(\widetilde{\lambda}_\ell) \text{ can be safely removed from the cache.}\end{aligned} \tag{9}$$

A definition of $\mathfrak{R}$ satisfying this property is derived as follows.

First, take any coherent template (e.g., $\widetilde{\boldsymbol{\lambda}'}_1$ in Fig. 3), and surround it by its metric ellipsoid at mismatch $\tilde{\mu}_{\mathrm{max}}$. Then, surround the metric ellipsoid in turn by its *bounding box*, the smallest coordinate box which contains the ellipsoid, e.g., Ref. [52]; the metric ellipse bounding box centered on $\widetilde{\boldsymbol{\lambda}'}_1$ is also shown in Fig. 3. Now, transform the bounding box into the semicoherent parameter space; practically, this may be achieved by expressing the coordinates of each vertex of the bounding box in the semicoherent metric coordinates. See Fig. 3 for the transformed bounding box of $\widetilde{\boldsymbol{\lambda}'}_1$ in the semicoherent parameter space, which is centered on $\widehat{\boldsymbol{\lambda}}'(\widetilde{\boldsymbol{\lambda}'}_1)$.

Note that, by definition, any semicoherent template $\widehat{\boldsymbol{\lambda}}$ outside of the transformed bounding box centered on $\widehat{\boldsymbol{\lambda}}'(\widehat{\boldsymbol{\lambda}'_1})$ *cannot* map to $\widetilde{\boldsymbol{\lambda}'}_1$ under $\widehat{\boldsymbol{\lambda}} \to \widetilde{\boldsymbol{\lambda}}_\ell(\widehat{\boldsymbol{\lambda}})$. Thus, to determine whether $\widetilde{\boldsymbol{\lambda}'}_1$ is accessible by $\widehat{\boldsymbol{\lambda}}$, we can compute whether $\widehat{\boldsymbol{\lambda}}$ is within the transformed bounding box of $\widehat{\boldsymbol{\lambda}}'(\widehat{\boldsymbol{\lambda}'}_1)$. To be conservative, however, we also surround $\widehat{\boldsymbol{\lambda}}$ by its bounding box as shown in Fig. 3 and instead compute whether the bounding boxes of $\widehat{\boldsymbol{\lambda}}'(\widehat{\boldsymbol{\lambda}'_1})$ and $\widehat{\boldsymbol{\lambda}}$ intersect.

To simplify the bounding box intersection calculation, we compare just the coordinates of the bounding boxes of $\widehat{\boldsymbol{\lambda}}'(\widehat{\boldsymbol{\lambda}'_1})$ and $\widehat{\boldsymbol{\lambda}}$ in *one* dimension; for reasons that will soon be apparent, we choose the lowest-dimensional coordinate, n_a. First, we define the relevance $\mathfrak{R}$ for both coherent and semicoherent templates:

$$\mathfrak{R}(\widetilde{\boldsymbol{\lambda}}_\ell) \equiv \text{the } \textit{maximum} \text{ value of } n_a \text{ within the transformed bounding box of } \widetilde{\boldsymbol{\lambda}}_\ell, \tag{10a}$$

and

$$\mathfrak{R}(\hat{\boldsymbol{\lambda}}) \equiv \text{the } \textit{minimum} \text{ value of } n_a \text{ within the bounding box of } \hat{\boldsymbol{\lambda}}. \tag{10b}$$

We now compute $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}'}_\ell)$ and $\mathfrak{R}(\widehat{\boldsymbol{\lambda}})$; in Fig. 3, $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}'}_\ell)$ is the n_a coordinate of the rightmost edge of the transformed bounding box of $\widehat{\boldsymbol{\lambda}}'(\widetilde{\boldsymbol{\lambda}'}_\ell)$, and $\mathfrak{R}(\widehat{\boldsymbol{\lambda}})$ is the n_a coordinate of the leftmost edge of the bounding box of $\widehat{\boldsymbol{\lambda}}$. In this example, $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}'}_\ell) < \mathfrak{R}(\widehat{\boldsymbol{\lambda}})$, and it follows from the definition of $\mathfrak{R}$ in Eqs. (10) that the bounding boxes of $\widehat{\boldsymbol{\lambda}}'(\widehat{\boldsymbol{\lambda}'}_1)$ and $\widehat{\boldsymbol{\lambda}}$ cannot intersect.

On the other hard, let us choose another coherent template $\widetilde{\boldsymbol{\lambda}''}_1$ and examine its relevance $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}'}_1)$; here, we have $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}''}_1) > \mathfrak{R}(\widehat{\boldsymbol{\lambda}})$ (see Fig. 3). From the simplified bounding box intersection calculation, we conclude that the bounding boxes of $\widehat{\boldsymbol{\lambda}}''(\widehat{\boldsymbol{\lambda}''}_1)$ and $\widehat{\boldsymbol{\lambda}}$ *could potentially* intersect, since at least in the n_a dimension the bounding boxes overlap (although in this example, the bounding boxes do not overlap in the n_b dimension).

Finally, if for some $\widehat{\boldsymbol{\lambda}}$ we have $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}'}_1) < \mathfrak{R}(\widehat{\boldsymbol{\lambda}})$, then this condition is guaranteed to remain true for all remaining $\widehat{\boldsymbol{\lambda}}$ in the template bank. This is simply a consequence of the algorithm used to generate the semicoherent template bank [52], which operates as follows: first, values of n_a are generated in a constant range $[n_{a\mathrm{min}}, n_{a\mathrm{max}}]$; then, for each value of n_a, values of n_b are generated in ranges $[n_b(n_a)_{\mathrm{min}}, n_b(n_a)_{\mathrm{max}}]$ dependent on n_a, and so on. It follows that the value of n_a can only increase during the generation of the semicoherent template bank, and since $\mathfrak{R}(\widehat{\boldsymbol{\lambda}})$ is defined in terms of n_a, it, too, can only increase.

To summarize, the relevance operator $\mathfrak{R}$ defined by Eqs. (10) satisfies the desired property given by Eq. (9). In Fig. 3, since $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}'}_1) < \mathfrak{R}(\widehat{\boldsymbol{\lambda}})$, the cache management algorithm would discard any coherent statistics associated with $\widetilde{\boldsymbol{\lambda}'}_1$ from memory, since they cannot be accessed by $\widehat{\boldsymbol{\lambda}}$ nor any remaining semicoherent template. On the other hard, the algorithm would retain any coherent statistics associated with $\widetilde{\boldsymbol{\lambda}''}_1$, since they could still be needed for future semicoherent templates; indeed, in Fig. 3, it is clear that the next semicoherent template in the bank, labeled $\widehat{\boldsymbol{\lambda}'''}$, could require coherent statistics associated with $\widetilde{\boldsymbol{\lambda}''_1}$, since the bounding boxes of $\widehat{\boldsymbol{\lambda}}''(\widehat{\boldsymbol{\lambda}''_1})$ and $\widehat{\boldsymbol{\lambda}'''}$ intersect.

The cache management algorithm described above is implemented in the main search loop in steps 7–11 (Fig. 1). In step 7, the cache is interrogated for a required $\mathcal{F}$-statistic value $2\mathcal{F}(\widetilde{\boldsymbol{\lambda}}_\ell)$: if it is in the cache, it is retrieved and utilized (step 8); otherwise, it is computed and inserted into the cache (step 9). In the latter case, the cache is also checked to see if any cache items can be discarded. Starting with step 10, cache items indexed by $\widetilde{\boldsymbol{\lambda}}_\ell$ are retrieved in order of ascending $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}}_\ell)$. If $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}}_\ell) < \mathfrak{R}(\widehat{\boldsymbol{\lambda}})$, the cache items are discarded (step 11). Only one cache item is removed at any one time, and therefore the memory usage of the cache will either remain constant or increase by one item per main search loop iteration. The cache is implemented using two data structures, e.g., Ref. [65]: a binary heap to rank cache items by $\mathfrak{R}(\widetilde{\boldsymbol{\lambda}}_\ell)$ and a hash table to find cache items indexed by $\widetilde{\boldsymbol{\lambda}}_\ell$.

IV. MODELS OF WEAVE BEHAVIOR

This section presents semianalytic models of the WEAVE implementation. It greatly facilitates the practical usage of any search method if its behavior can be characterized *a priori* as much as possible using a computationally-cheap model. For example, a model of the distribution of $\mathcal{F}$-statistic mismatches (Sec. IV A) permits the estimation of the sensitivity of a particular search setup [25], which in turn allows the setup to be optimized so as to maximize sensitivity [24]. Similarly, models of the number of coherent and semicoherent templates (Sec. IV B) and computational cost (Sec. IV C) allow the parameters of

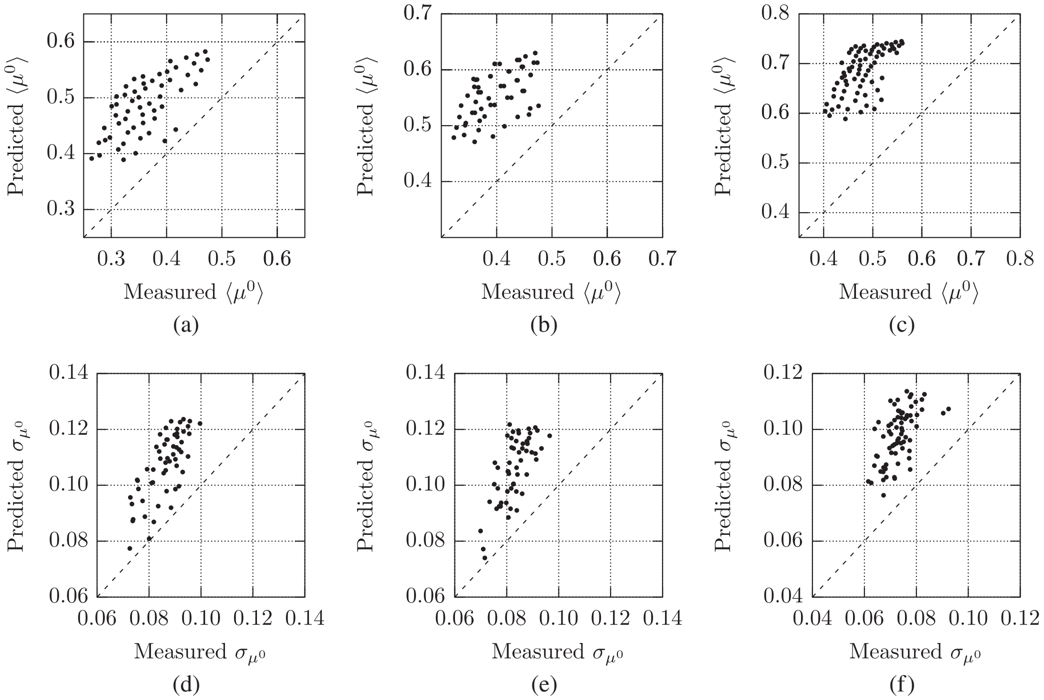


FIG. 4. Predicted means (top row) and standard deviations (bottom row) of the distributions of $\mathcal{F}$-statistic mismatches, against their measured values, for (left column to right column) the three search setups listed in Table I. The dotted line denotes equality between predicted and measured values.

the optimal search setup to be estimated [24]. The memory usage (Sec. IV D) and input data bandwidth (Sec. IV E) required by the implementation are also important properties when implementing a search pipeline.

Each model presented in this section is implemented as an OCTAVE [66] script and is freely available as part of the OctApps [67] script library.

A. Distribution of $\mathcal{F}$-statistic mismatches

The distribution of the mismatch between the $\mathcal{F}$-statistic computed at an exact signal location, and at the nearest point in the WEAVE semicoherent template bank, gives an idea of the expected loss in signal-to-noise ratio due to the necessary coarseness of the template bank. Figure 4 plots the predicted means and standard deviations of WEAVE $\mathcal{F}$-statistic mismatch distributions, against their measured values, for a variety of setups given in Table I. The distributions were measured using software injection studies, where relatively strong ($h_0/\sqrt{S_h} \gtrsim 70\ \mathrm{Hz}^{1/2}$) simulated signals are added to Gaussian-distributed noise and then searched for using LALAPPS_WEAVE.

TABLE I. Details of search setups used to test model in Sec. IV A. Columns are (left to right): number of segments, timespan of each segment in hours, total time span of all segments in days, number of $(\tilde{\mu}_{\max}, \hat{\mu}_{\max})$ pairs used, ranges of maximum coherent and semicoherent mismatches, and average number of injections per $(\tilde{\mu}_{\max}, \hat{\mu}_{\max})$ pair.

N	$\tilde{T}$	$\hat{T}$	$\lvert\{(\tilde{\mu}_{\max}, \hat{\mu}_{\max})\}\rvert$	$\tilde{\mu}_{\max}$	$\hat{\mu}_{\max}$	n_{inj}
228	25.0	256.0	56	0.1–1.2	1.5–12.0	111.4
195	30.0	256.2	56	0.1–1.6	2.0–12.0	111.4
147	40.0	256.6	71	0.1–1.5	4.0–24.0	111.4

The predicted means and standard deviations are from the model presented in Ref. [47] and are generally conservative: Fig. 4 shows that the model generally overestimates the mean $\mathcal{F}$-statistic mismatch by ~0.13 [Fig. 4(a)] to ~0.20 [Fig. 4(c)], and the predicted standard deviations imply slightly broader distributions than were measured. As explored in Ref. [47], the relationship between the maximum mismatches of the coherent and semicoherent template banks (which are inputs to LALAPPS_WEAVE) and the $\mathcal{F}$-statistic mismatch distribution (which is output by LALAPPS_WEAVE) is difficult to model when the former are large, e.g., $\gtrsim 1$.

In addition, an optimization implemented in WEAVE but not accounted for in the model of Ref. [47] complicates the picture: the coherent and semicoherent template banks are constructed to have equally-spaced templates in the frequency parameter f. This permits (in step 9 of Fig. 1) the simultaneous computation of a series of $2\mathcal{F}$ values at equally-spaced values of f across the frequency parameter space, which can be performed efficiently using fast Fourier transform–based algorithms (see Sec. IV C). The construction of equal-frequency-spacing coherent and semicoherent template banks is performed by first constructing each bank independently and then reducing the frequency spacing in all banks to that of the smallest frequency spacing in any bank. This construction will always reduce the maximum possible mismatch in each grid, but never increase it, and so we would expect the mean $\mathcal{F}$-statistic mismatch measured by WEAVE to be smaller than that predicted by the model of Ref. [47].

The model of Ref. [47] is implemented in the OCTAPPS script WEAVEFSTATMISMATCH.M.

B. Number of templates

Since the WEAVE coherent and semicoherent template banks are constructed using lattices (see Sec. II C), the number of templates in each is estimated starting from the formula, e.g., Refs. [51,52],

$$\mathcal{N} = \theta \mu_{\max}^{-n/2} \sqrt{\det \mathbf{g}}\, \mathcal{V}, \tag{11}$$

where $\mathcal{V}$ is the volume of the n-dimensional parameter space, $\mathbf{g}$ is the parameter-space metric, and $\mu_{\max}$ is the maximum mismatch. The *normalized thickness* θ is a property of the particular lattice used to generate the template bank, e.g., Ref. [48].

The parameter-space volume is given explicitly by the following expressions:

$$\mathcal{V} = \mathcal{V}_{n_a n_b} \prod_{s=0}^{s_{\max}} \mathcal{V}_{f^{(s)}}, \tag{12}$$

$$\begin{aligned} \mathcal{V}_{n_a n_b} &= \max\{\beta_a \beta_b, \mathcal{V}'_{n_a n_b}\}, \\ \mathcal{V}'_{n_a n_b} &= 2(\pi + 4\beta_b + \beta_a \beta_b) \begin{cases} \frac{(\alpha_{\max} - \alpha_{\min})(\sin\delta_{\max} - \sin\delta_{\min})}{4\pi} & \text{or} \\ \frac{1}{K}, \end{cases} \end{aligned} \tag{13}$$

$$\mathcal{V}_{f^{(s)}} = \beta_{f^{(s)}} + f_{\max}^{(s)} - f_{\min}^{(s)}. \tag{14}$$

Here, $\boldsymbol{\beta}$ is the vector of which the components are the extents of the bounding box of $\mathbf{g}$ in each dimension; it is used to ensure that the volume of the parameter space in each dimension is not smaller than the extent of a single template. In Eq. (13), the volume of the sky parameter space may be specified either by a rectangular patch $[\alpha_{\min}, \alpha_{\max}] \otimes [\delta_{\min}, \delta_{\max}]$ or by the number K of equal-size sky patches (see Sec. III A).

Finally, the total number of coherent and semicoherent templates, $\tilde{\mathcal{N}}$ and $\widehat{\mathcal{N}}$, respectively, are given by

$$\tilde{\mathcal{N}} = 1.436 \sum_{\ell=0}^{N-1} \theta \tilde{\mu}_{\max}^{-n/2} \sqrt{\det \tilde{\mathbf{g}}_{\ell}}\, \mathcal{V}, \tag{15}$$

$$\widehat{\mathcal{N}} = \theta \widehat{\mu}_{\max}^{-n/2} \sqrt{\det \widehat{\mathbf{g}}}\, \widehat{\mathcal{V}}. \tag{16}$$

The numerical prefactor on the right-hand side of Eq. (15) is chosen to better match $\tilde{\mathcal{N}}$ to the number of coherent templates actually computed by LALAPPS_WEAVE: the coherent parameter space is augmented with additional padding along its boundaries to ensure that it encloses the semicoherent parameter space, i.e., that it includes a nearest neighbor for every $\widehat{\boldsymbol{\lambda}}$.

Equations (15) and (16) are used to predict the number of templates computed by LALAPPS_WEAVE for a variety of search setups detailed in Table II. Figure 5 plots the predicted $\tilde{\mathcal{N}}$ and $\widehat{\mathcal{N}}$ against the values measured by running LALAPPS_WEAVE. Reasonable agreement is achieved between predicted and measured $\tilde{\mathcal{N}}$ [Fig. 5(a)]; while Eq. (15) sometimes underestimates the number of coherent templates, it rarely does so by more than a factor of a few. Better agreement is seen between predicted and measured $\widehat{\mathcal{N}}$ [Fig. 5(b)].

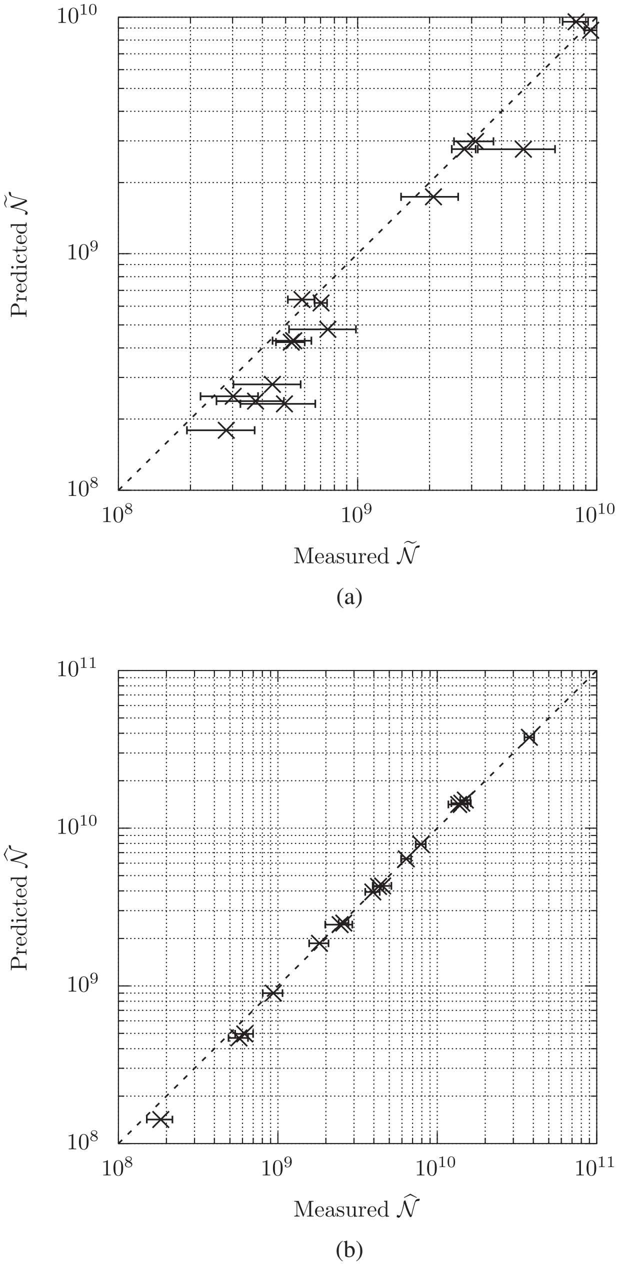
Equations (15) and (16) are implemented in the OCTAPPS script WEAVETEMPLATECOUNT.M.

C. Computational cost

The total computational cost C_{tot} of a particular search setup may be modeled in terms of the number of coherent

TABLE II. Details of search setups used to test models in Secs. IV B–IV E. Columns are (left to right): number of segments, time span of each segment in hours, total time span of all segments in days, maximum coherent and semicoherent mismatches, number of patches used to partition sky, number of sky patches used to test models, range of frequency parameter space in Hz, range of spin-down parameter space in Hz/s, and $\mathcal{F}$-statistic algorithm (D = demodulation, R = resampling).

N	$\tilde{T}$	$\hat{T}$	$\tilde{\mu}_{\max}$	$\hat{\mu}_{\max}$	K	$\lvert\{k\}\rvert$	Δf	$\Delta\dot{f}$	$\mathcal{F}$alg.
10	25.0	10.5	0.1	0.1	100	24	0.1	4.5×10^{-9}	R
10	25.0	10.5	0.1	0.1	100	24	0.5	1.0×10^{-9}	R
10	25.0	10.5	0.1	0.1	100	24	0.1	5.0×10^{-9}	D
10	25.0	10.5	0.1	0.2	24	24	0.1	3.6×10^{-9}	R
10	25.0	10.5	0.1	0.2	24	24	0.5	8.0×10^{-10}	R
10	25.0	10.5	0.1	0.2	24	24	0.1	4.0×10^{-9}	D
29	25.0	33.8	0.1	0.5	100	24	0.1	8.1×10^{-10}	R
29	25.0	33.8	0.1	0.5	100	24	0.5	1.8×10^{-10}	R
29	25.0	33.8	0.1	0.5	100	24	0.1	9.0×10^{-10}	D
29	25.0	33.8	0.1	0.5	500	24	0.1	9.0×10^{-10}	R
29	25.0	33.8	0.1	0.5	500	24	0.5	2.0×10^{-10}	R
29	25.0	33.8	0.1	0.5	500	24	0.1	1.0×10^{-9}	D
90	25.0	105.2	0.01	0.8	2700	24	0.1	1.8×10^{-10}	R
90	25.0	105.2	0.01	0.8	2700	24	0.5	4.0×10^{-11}	R
90	25.0	105.2	0.01	0.8	2700	24	0.1	2.0×10^{-10}	D
90	25.0	105.2	0.1	0.8	1000	24	0.1	2.7×10^{-9}	R
90	25.0	105.2	0.1	0.8	1000	24	0.5	6.0×10^{-10}	R
90	25.0	105.2	0.1	0.8	1000	24	0.1	3.0×10^{-9}	D
228	25.0	256.0	0.1	2	5000	24	0.1	8.1×10^{-10}	R
228	25.0	256.0	0.1	2	5000	24	0.5	1.8×10^{-10}	R
228	25.0	256.0	0.1	2	5000	24	0.1	9.0×10^{-10}	D
228	25.0	256.0	0.1	2	5000	24	0.1	5.4×10^{-10}	R
228	25.0	256.0	0.1	2	5000	24	0.5	1.2×10^{-10}	R
228	25.0	256.0	0.1	2	5000	24	0.1	6.0×10^{-10}	D



(a)

(b)

FIG. 5. Predicted number of (a) coherent templates [Eq. (15)] and (b) semicoherent templates [Eq. (16)], against their measured values. The error bars denote the standard deviations of measured $\widetilde{\mathcal{N}}$ and $\widehat{\mathcal{N}}$ when averaged over different sky patches. The dotted line denotes equality between predicted and measured values. The search setups used in this figure are listed in Table II.

$\widetilde{\mathcal{N}}$ and semicoherent $\widehat{\mathcal{N}}$ templates (see Sec. IV B), the number of segments N, and the number of detectors N^{det}. Following Ref. [24], we write

$$C_{\mathrm{tot}}(\widetilde{\mathcal{N}},\widehat{\mathcal{N}},N,N^{\mathrm{det}}) = \widetilde{C}(\widetilde{\mathcal{N}},N^{\mathrm{det}}) + \widehat{C}(\widehat{\mathcal{N}},N,N^{\mathrm{det}}) + C_{\mathrm{other}}, \tag{17}$$

where $\widetilde{C}$ and $\widehat{C}$ denote the computational cost of the coherent and semicoherent stages of the search method, respectively, and C_{other} denotes any unmodeled computational costs.

The computational cost model takes as input *fundamental timing constants* which give the time taken to complete certain fundamental computations. Their values are highly dependent on various properties of the computer hardware used to run LALAPPS_WEAVE, such as the processor speed and cache sizes, as well as what other programs were using the computer hardware at the same time as LALAPPS_WEAVE. Some values are also specific to the search setups detailed in Table II. For the interest of the reader, Table III lists representative values of the fundamental timing constants obtained on a particular computer cluster.

The coherent cost $\widetilde{C}$ is simply the cost of computing the $\mathcal{F}$-statistic (step 9 of Fig. 1):

$$\widetilde{C}(\widetilde{\mathcal{N}},N^{\mathrm{det}}) = \widetilde{\mathcal{N}} N^{\mathrm{det}} \tau_{\mathcal{F}}^{\mathrm{eff}}(\Delta f,\widetilde{T},\mathcal{F}\,\mathrm{alg}). \tag{18}$$

The fundamental timing constant $\tau_{\mathcal{F}}^{\mathrm{eff}}$ gives the time taken to compute the $\mathcal{F}$-statistic per template and per detector and is further described in Ref. [68]. Its value depends primarily upon the range of the frequency parameter space Δf, the coherent segment length $\widetilde{T}$, and the algorithm used to compute the $\mathcal{F}$-statistic. Choices for the latter are the *resampling* algorithm, e.g., Refs. [19,69], which computes the $\mathcal{F}$-statistic over a wide band of frequencies efficiently using the fast Fourier transform and is generally used to performing an initial wide-parameter-space search, and the *demodulation* algorithm of Ref. [70], which uses a Dirichlet kernel to compute the $\mathcal{F}$-statistic more efficiently at a single frequency or over a narrow frequency band and is therefore used to perform follow-up searches of localized parameter spaces around interesting candidates. The additional cost of managing the cache of computed $\mathcal{F}$-statistic values (steps 8, 10, and 11) is amortized into $\widetilde{C}$.

TABLE III. Representative values of the fundamental timing constants of the computational cost model detailed in Sec. IV C. These values were computed by running LALAPPS_WEAVE on a computer cluster of Intel Xeon E5-2658V4 processors running at 2.30 GHz. Some values are specific to the search setups detailed in Table II.

	Representative value/s	
Fundamental timing constant	Demodulation	Resampling
$\tau_{\mathcal{F}}^{\mathrm{eff}}$	2.5×10^{-6}	$(1.5\text{–}4)\times10^{-7}$
τ_{iter}	1.4×10^{-10}	
τ_{query}	8.6×10^{-11}	
$\tau_{2\mathcal{F}_{\mathrm{mean}}}$	8.3×10^{-10}	
$\tau_{2\mathcal{F}_{\mathrm{sum}}}$	7.3×10^{-10}	
$\tau_{\mathcal{B}_{\mathrm{SGL}}}$	9.9×10^{-9}	
τ_{out}	7.9×10^{-10}	

The semicoherent cost

$$\begin{aligned}\widehat{C}(\widehat{\mathcal{N}}, N, N^{\mathrm{det}}) = {}& \widehat{C}_{\mathrm{iter}}(\widehat{\mathcal{N}}) + \widehat{C}_{\mathrm{query}}(\widehat{\mathcal{N}}, N) \\ & + \widehat{C}_{2\mathcal{F}_{\mathrm{sum}}}(\widehat{\mathcal{N}}, N, N^{\mathrm{det}}) + \widehat{C}_{2\mathcal{F}_{\mathrm{mean}}}(\widehat{\mathcal{N}}) \\ & + \widehat{C}_{\widehat{\mathcal{B}}_{\mathrm{SGL}}}(\widehat{\mathcal{N}}) + \widehat{C}_{\mathrm{out}}(\widehat{\mathcal{N}}) \end{aligned} \tag{19}$$

has a number of components:

(i) $\widehat{C}_{\mathrm{iter}}$ is the cost of iterating over the semicoherent template bank (steps 5 and 16 of Fig. 1);

(ii) $\widehat{C}_{\mathrm{query}}$ is the cost of finding the nearest templates in the coherent template banks (step 6 and 13) and of interrogating the cache of computed $\mathcal{F}$-statistic values (step 7);

(iii) $\widehat{C}_{2\mathcal{F}_{\mathrm{sum}}}$ is the cost of computing $2\mathcal{F}_{\mathrm{sum}}$ and, if required, $2\mathcal{F}^{X}_{\mathrm{sum}}$ (step 12);

(iv) $\widehat{C}_{2\mathcal{F}_{\mathrm{mean}}}$ is the cost of computing $2\mathcal{F}_{\mathrm{mean}}$ (step 14);

(v) $\widehat{C}_{\widehat{\mathcal{B}}_{\mathrm{SGL}}}$ is the cost of computing $\widehat{\mathcal{B}}_{\mathrm{SGL}}$, if required (step 14); and

(vi) $\widehat{C}_{\mathrm{out}}$ is the cost of adding candidates to top lists (step 15).

These components of $\widehat{C}$ are further defined in terms of $\widehat{\mathcal{N}}$, N, N^{det}, and various fundamental timing constants (see Table III) as follows:

$$\widehat{C}_{\mathrm{iter}}(\widehat{\mathcal{N}}) = \widehat{\mathcal{N}}\tau_{\mathrm{iter}}; \tag{20}$$

$$\widehat{C}_{\mathrm{query}}(\widehat{\mathcal{N}}, N) = \widehat{\mathcal{N}} N \tau_{\mathrm{query}}; \tag{21}$$

$$\begin{aligned}\widehat{C}_{2\mathcal{F}_{\mathrm{sum}}}(\widehat{\mathcal{N}}, N, N^{\mathrm{det}}) = {}& \widehat{\mathcal{N}}(N-1)\tau_{2\mathcal{F}_{\mathrm{sum}}} \\ & \times \begin{cases} 1 + N^{\mathrm{det}} & \text{if } 2\mathcal{F}^{X}_{\mathrm{sum}} \text{ reqd,} \\ 1 & \text{otherwise;} \end{cases}\end{aligned} \tag{22}$$

$$\widehat{C}_{2\mathcal{F}_{\mathrm{mean}}}(\widehat{\mathcal{N}}) = \widehat{\mathcal{N}}\tau_{2\mathcal{F}_{\mathrm{mean}}}; \tag{23}$$

$$\widehat{C}_{\widehat{\mathcal{B}}_{\mathrm{SGL}}}(\widehat{\mathcal{N}}) = \widehat{\mathcal{N}}\tau_{\widehat{\mathcal{B}}_{\mathrm{SGL}}}; \tag{24}$$

$$\widehat{C}_{\mathrm{out}}(\widehat{\mathcal{N}}) = \widehat{\mathcal{N}}\tau_{\mathrm{out}} \times \text{number of top lists.} \tag{25}$$

Figure 6 compares the computational cost model of Eqs. (17)–(20) against the measured computational cost of LALAPPS_WEAVE (see Table III), using the search setups detailed in Table II. The total computational cost of LALAPPS_WEAVE is generally well modeled [Fig. 6(a)], and the unmodeled component of the measured computational cost is low [Fig. 6(b)]. The coherent computational cost $\tilde{C}$ of Eq. (18) and the components of the semicoherent cost $\widehat{C}$ of Eq. (19) are also in good agreement [Figs. 6(c)–6(i)].

Equations (17)–(20) are implemented in the OCTAPPS script WEAVERUNTIME.M.

D. Memory usage

The memory usage M of LALAPPS_WEAVE is modeled by

$$M = M_{\mathcal{F}} + M_{\mathrm{cache}}. \tag{26}$$

The first term on the right-hand side, $M_{\mathcal{F}}$, is the memory usage of the $\mathcal{F}$-statistic algorithm (which includes the gravitational-wave detector data) and is further described in Ref. [68]. The second term, M_{cache}, is the memory usage of the cache of computed $\mathcal{F}$-statistic values and is further given by

$$M_{\mathrm{cache}} = N^{\mathrm{max}}_{\mathrm{cache}} m_{2\mathcal{F}} \begin{cases} 1 + N^{\mathrm{det}} & \text{if } 2\mathcal{F}^{X}_{\mathrm{sum}} \text{required,} \\ 1 & \text{otherwise,} \end{cases} \tag{27}$$

where $N^{\mathrm{max}}_{\mathrm{cache}}$ is the maximum size of the cache (across all segments) and $m_{2\mathcal{F}} \equiv 4 \times 2^{-20}$ mebibytes is the memory required to store one $2\mathcal{F}$ value as a 4 byte single precision floating-point number. The maximum cache $N^{\mathrm{max}}_{\mathrm{cache}}$ cannot easily be predicted from first principles, i.e., given the search setup, parameter space, and other input arguments to LALAPPS_WEAVE. Instead, it is measured by running LALAPPS_WEAVE in a special mode which simulates the performance of the cache but without computing any $\mathcal{F}$-statistic or derived values; essentially, it follows Fig. 1 but with the first part of steps 9, 12, and 14 omitted.

Figure 7 plots the predicted memory usage of Eqs. (26) and (27) against the measured memory usage of LALAPPS_WEAVE, using the search setups detailed in Table II. The $\mathcal{F}$-statistic is computed using both the resampling and demodulation algorithms: in the former case, both $\mathcal{F}_{\mathrm{mean}}$ and $\widehat{\mathcal{B}}_{\mathrm{SGL}}$ are computed, thereby triggering the first case in Eq. (27); in the latter case, only $\mathcal{F}_{\mathrm{mean}}$ is computed, thereby triggering the second case in Eq. (27). Good agreement between predicted and measured memory usage is seen for both algorithms.

Equations (26) and (27) are also implemented in the OCTAPPS script WEAVERUNTIME.M.

E. Input data bandwidth

Our final WEAVE model concerns what bandwidth of the input gravitational-wave detector data is required to search a given frequency range. For most continuous-wave search pipelines, short (typically 1800 s) contiguous segments of gravitational-wave strain data are Fourier transformed, and the resulting complex spectra stored as short Fourier transform (SFT) files. A continuous-wave search of a large frequency parameter space will generally be divided into smaller jobs, with each job searching a smaller partition of the whole frequency parameter space. Each job therefore requires that only a small bandwidth out of the full SFT spectra be read into memory.

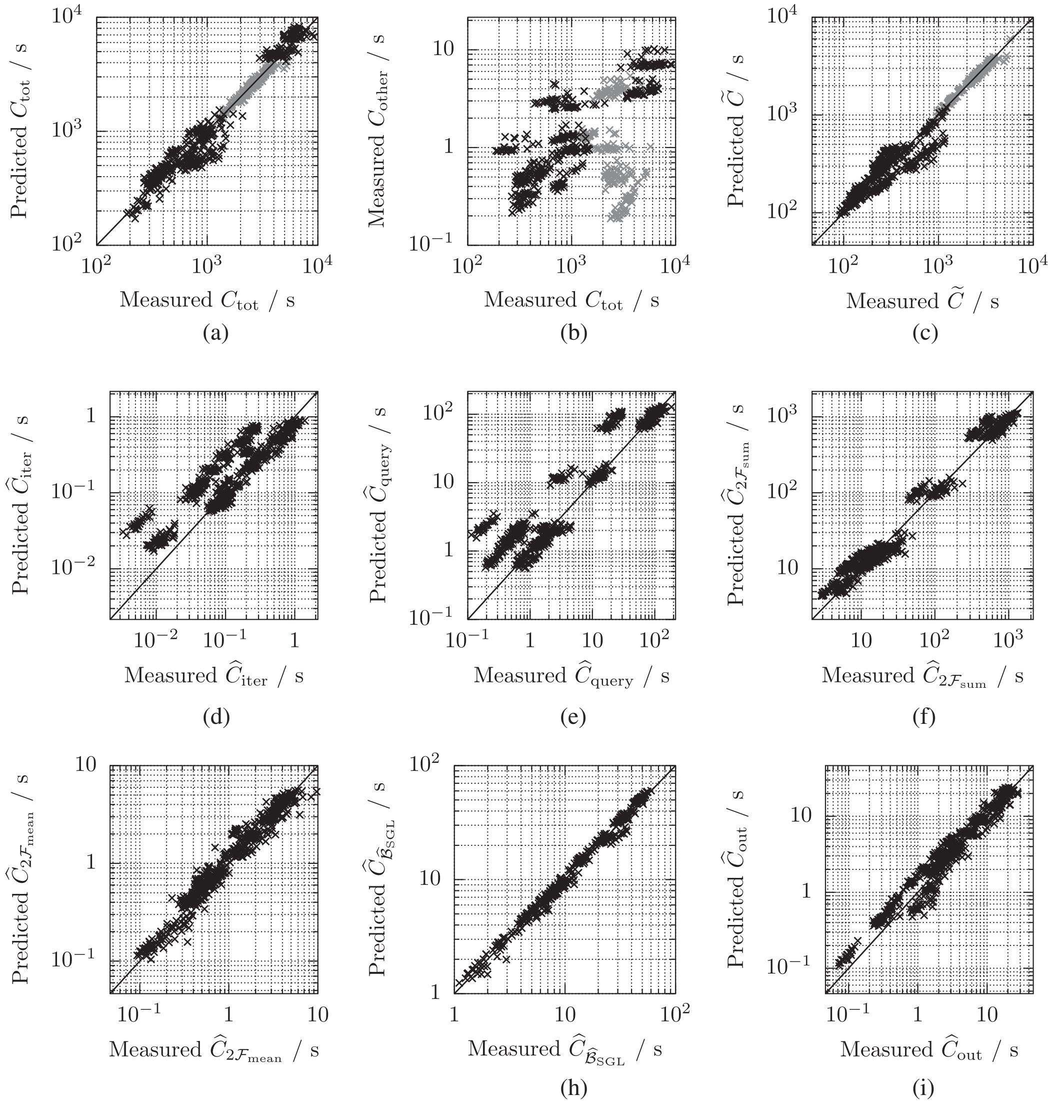


FIG. 6. (a) Predicted vs measured total computational cost C_{tot} [Eq. (17)]. (b) Measured unmodeled computational cost C_{other} [Eq. (17)] vs measured total computational cost C_{tot}. (c) –(i) Predicted vs measured values of the coherent computational cost $\tilde{C}$ of Eq. (18) and of the components of the semicoherent computational cost $\hat{C}$ of Eq. (19). In (a)–(c), grey crosses and black crosses are used to distinguish values computed using the demodulation and resampling $\mathcal{F}$-statistic algorithms, respectively; otherwise, black crosses are used for values which are independent of the choice of $\mathcal{F}$-statistic algorithm. The search setups used in this figure are listed in Table II.

Given an input frequency parameter space $[f_{\min}, f_{\max}]$ and spin-down parameter space $[\dot{f}_{\min}, \dot{f}_{\max}]$, predicting the bandwidth of the SFT spectra required by LALAPPS_WEAVE proceeds in several steps. First, the input parameter spaces are augmented to account for the extra padding of the WEAVE template banks,

$$f'_{\min} = (1 - x_{\mathrm{pad}})f_{\min}, \qquad f'_{\max} = (1 + x_{\mathrm{pad}})f_{\max}, \quad (28a)$$

$$\dot{f}'_{\min} = (1 - \dot{x}_{\mathrm{pad}})\dot{f}_{\min}, \qquad \dot{f}'_{\max} = (1 + \dot{x}_{\mathrm{pad}})\dot{f}_{\max}, \quad (28b)$$

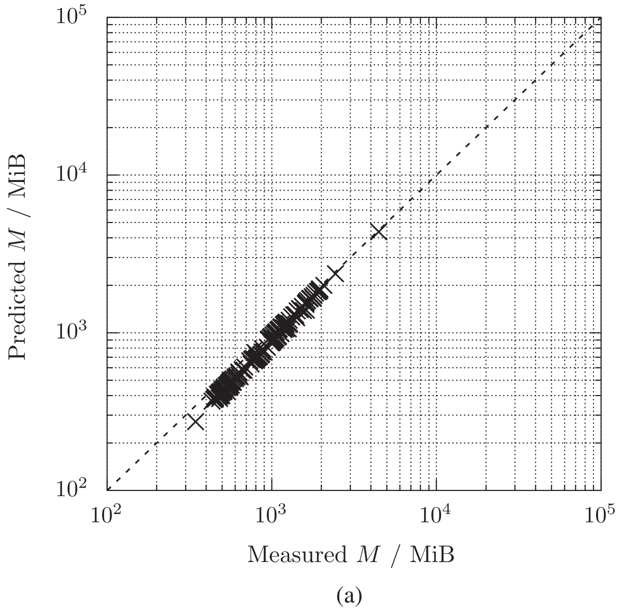
where $x_{\mathrm{pad}} \equiv 10^{-3}$ and $\dot{x}_{\mathrm{pad}} \equiv 10^{-10}$ are empirically chosen. Next, the maximum frequency range $[f''_{\min}, f''_{\max}]$ is found by evolving the frequency spin-down parameter space $[f'_{\min}, f'_{\max}] \otimes [\dot{f}'_{\min}, \dot{f}'_{\max}]$ from the reference time t_0 to the start and end times of each segment, t_ℓ^{start} and t_ℓ^{stop}, respectively:

$$f''_{\min} = \min\{f'_{\min} + \dot{f}'_{\min}(t - t_0) | t \in \{t_\ell^{\mathrm{start}}, t_\ell^{\mathrm{stop}}\}_{\ell=0}^{N-1}\}, \quad (29a)$$

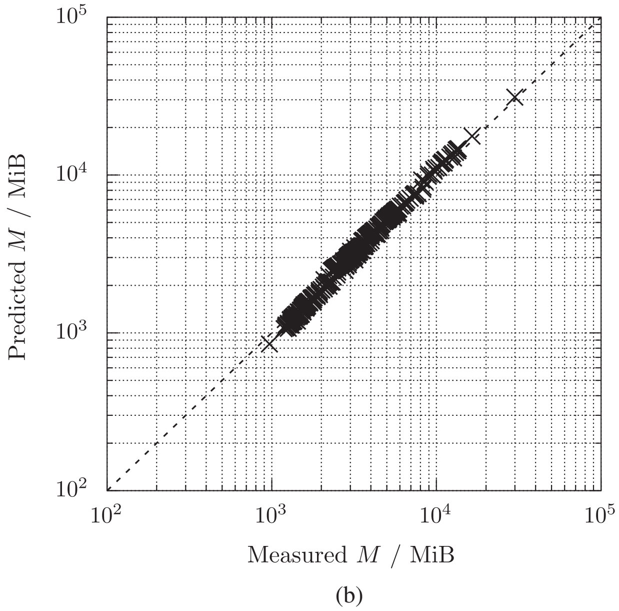
$$f''_{\max} = \max\{f'_{\max} + \dot{f}'_{\max}(t - t_0) | t \in \{t_\ell^{\mathrm{start}}, t_\ell^{\mathrm{stop}}\}_{\ell=0}^{N-1}\}. \quad (29b)$$

Finally, the SFT bandwidth $[f_{\min}^{\mathrm{SFT}}, f_{\max}^{\mathrm{SFT}}]$ of the SFT spectra which is required by LALAPPS_WEAVE is given by

$$f_{\min}^{\mathrm{SFT}} = (1 - x_{\mathrm{sky}})f''_{\min} - f_{\mathcal{F}}, \quad (30a)$$



(a)



(b)

FIG. 7. Predicted memory usage of LALAPPS_WEAVE, against its measured memory usage, when using the (a) demodulation and (b) resampling $\mathcal{F}$-statistic algorithms. The search setups used in this figure are listed in Table II.

$$f_{\max}^{\rm SFT} = (1 + x_{\rm sky}) f''_{\max} + f_{\mathcal{F}}. \tag{30b}$$

The $x_{\rm sky}$ enlarges $[f''_{\min}, f''_{\max}]$ to account for the maximum frequency-dependent Doppler modulation of a continuous-wave signal due to the sidereal and orbital motions of the Earth and is given by

$$x_{\rm sky} = \frac{2\pi}{c}\left(\frac{D_{\rm ES}}{1\text{ year}} + \frac{R_{\rm E}}{1\text{ day}}\right), \tag{31}$$

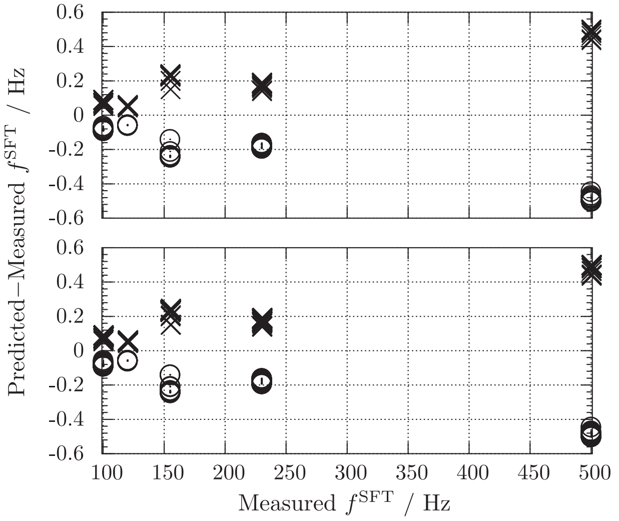


FIG. 8. Difference between predicted and measured $f_{\min}^{\rm SFT}$ (circles) and $f_{\max}^{\rm SFT}$ (crosses), against the measured $f_{\min}^{\rm SFT}$ and $f_{\max}^{\rm SFT}$, respectively, for the demodulation (top plot) and resampling (bottom plot) $\mathcal{F}$-statistic algorithms. The search setups used in this figure are listed in Table II.

where c is the speed of light, $D_{\rm ES}$ is the Earth-Sun distance, and $R_{\rm E}$ is the radius of the Earth. Additional padding of $[f''_{\min}, f''_{\max}]$ is also required for use by the chosen $\mathcal{F}$-statistic algorithm and is given by $f_{\mathcal{F}}$; see Ref. [68].

Figure 8 compares the model of Eqs. (28)–(31) against the behavior of LALAPPS_WEAVE when run with the search setups detailed in Table II. Note that the model satisfies

$$\text{predicted}\, f_{\min}^{\rm SFT} - \text{measured} f_{\min}^{\rm SFT} \le 0,$$

i.e., all circles plotted in Fig. 8 are below the horizontal axis, and

$$\text{predicted}\, f_{\max}^{\rm SFT} - \text{measured} f_{\max}^{\rm SFT} \ge 0,$$

i.e., all crosses plotted in Fig. 8 are above the horizontal axis. The model is therefore conservative; i.e., it may predict a slightly-larger SFT bandwidth than required but should never predict a smaller SFT bandwidth, which would cause a fatal error in LALAPPS_WEAVE. The model is generally more conservative at higher frequencies, where the Doppler modulation due to the Earth's motion is higher.

Equations (28)–(31) are implemented in the OCTAPPS script WEAVEINPUTSFTBAND.M.

V. DISCUSSION

This paper details the WEAVE implementation of a semicoherent search method for continuous gravitational waves. It focuses on all-sky surveys for isolated continuous-wave sources, for which the parameter space is the sky position and frequency evolution of the source. We note, however, that the implementation is in fact indifferent to the parameter space being searched, as long as the relevant constant parameter-space metric is available. The implementation could therefore be adapted to search other parameter spaces for continuous-wave sources such as known low-mass x-ray binaries, for which the parameter space includes the evolution parameters of the binary orbit, using the metric of Ref. [71].

There is scope to improve the semianalytic models of the behavior of LALAPPS_WEAVE presented in Sec. IV. In particular, a more accurate model of the distribution of $\mathcal{F}$-statistic mismatches than that presented in Sec. IV A would allow the sensitivity of a search to be more accurately estimated without resorting to software injection studies. The memory model of Sec. IV D would also be improved if the maximum cache size $N^{\max}_{\mathrm{cache}}$ could be predicted from first principles.

In a forthcoming paper [72], we plan to more fully characterize the performance of the WEAVE implementation and compare it to an implementation of the method of Refs. [37,38] using a mock data challenge.

ACKNOWLEDGMENTS

We thank Bruce Allen and Heinz-Bernd Eggenstein for valuable discussions. K. W. is supported by ARC Grant No. CE170100004. Numerical simulations were performed on the Atlas computer cluster of the Max Planck Institute for Gravitational Physics.

APPENDIX: PROPERTIES OF EQUAL-AREA SKY PATCHES

The search program LALAPPS_WEAVE allows the sky search parameter space to be partitioned into K patches and a patch selected by an index k. Tests of this feature found that, provided $K \ll \mathcal{N}^1_1$ (the number of templates with just one patch), the variation in the number of templates between patches $\Delta\mathcal{N}_K$ is generally small and well approximated by

$$\Delta\mathcal{N}_K = \begin{cases} 1.48\times10^{-2} + 5.35\times10^{-4}K & K \le 100, \\ 8.48\times10^{-2} + 4.03\times10^{-5}K & K > 100. \end{cases} \tag{A1}$$

The ratio $\sum_{k=0}^{K-1} \mathcal{N}^k_K / \mathcal{N}^1_1$ of the number of templates in all K patches to the number of templates with just one patch is generally $\lesssim 7\%$. The union of all templates in a set of K patches also faithfully reproduces the unpartitioned template bank, i.e., with just one patch.

[1] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Observation of Gravitational Waves from a Binary Black Hole Merger, Phys. Rev. Lett. **116**, 061102 (2016).

[2] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW151226: Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence, Phys. Rev. Lett. **116**, 241103 (2016).

[3] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW170104: Observation of a 50-Solar-Mass Binary Black Hole Coalescence at Redshift 0.2, Phys. Rev. Lett. **118**, 221101 (2017).

[4] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW170608: Observation of a 19 solar-mass binary black hole coalescence, Astrophys. J. Lett. **851**, L35 (2017).

[5] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW170814: A Three-Detector Observation of Gravitational Waves from a Binary Black Hole Coalescence, Phys. Rev. Lett. **119**, 141101 (2017).

[6] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, Phys. Rev. Lett. **119**, 161101 (2017).

[7] B. P. Abbott *et al.* (LIGO Scientific Collaboration), LIGO: The Laser Interferometer Gravitational-Wave Observatory, Rep. Prog. Phys. **72**, 076901 (2009).

[8] J. Aasi *et al.* (LIGO Scientific Collaboration), Advanced LIGO, Classical Quantum Gravity **32**, 074001 (2015).

[9] F. Acernese *et al.* (Virgo Collaboration), Advanced Virgo: A second-generation interferometric gravitational wave detector, Classical Quantum Gravity **32**, 024001 (2015).

[10] K. Somiya (KAGRA Collaboration), Detector configuration of KAGRA-the Japanese cryogenic gravitational-wave detector, Classical Quantum Gravity **29**, 124007 (2012).

[11] C. S. Unnikrishnan, IndIGO and Ligo-India scope and plans for gravitational wave research and precision metrology in India, Int. J. Mod. Phys. D **22**, 1341010 (2013).

[12] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), First search for gravitational waves from known pulsars with Advanced LIGO, Astrophys. J. **839**, 12 (2017).

[13] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Search for gravitational waves from Scorpius X-1 in the first Advanced LIGO observing run with a hidden Markov model, Phys. Rev. D **95**, 122003 (2017).

[14] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), All-sky search for periodic gravitational waves in the O1 LIGO data, Phys. Rev. D **96**, 062002 (2017).

[15] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Upper limits on gravitational waves from Scorpius X-1 from a model-based cross-correlation search in Advanced LIGO data, Astrophys. J. **847**, 47 (2017).

[16] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaboration), First low-frequency Einstein@Home all-sky search for continuous gravitational waves in Advanced LIGO data, Phys. Rev. D **96**, 122004 (2017).

[17] N. K. Johnson-McDaniel and B. J. Owen, Maximum elastic deformations of relativistic stars, Phys. Rev. D **88**, 044004 (2013).

[18] P. R. Brady, T. Creighton, C. Cutler, and B. F. Schutz, Searching for periodic sources with LIGO, Phys. Rev. D **57**, 2101 (1998).

[19] P. Jaranowski, A. Królak, and B. F. Schutz, Data analysis of gravitational-wave signals from spinning neutron stars: The signal and its detection, Phys. Rev. D **58**, 063001 (1998).

[20] K. Wette *et al.*, Searching for gravitational waves from Cassiopeia A with LIGO, Classical Quantum Gravity **25**, 235011 (2008).

[21] P. R. Brady and T. Creighton, Searching for periodic sources with LIGO. II. Hierarchical searches, Phys. Rev. D **61**, 082001 (2000).

[22] B. Abbott *et al.* (LIGO Scientific Collaboration), Einstein@Home search for periodic gravitational waves in LIGO S4 data, Phys. Rev. D **79**, 022001 (2009).

[23] G. Poghosyan, S. Matta, A. Streit, M. Bejger, and A. Królak, Architecture, implementation and parallelization of the software to search for periodic gravitational wave signals, Comput. Phys. Commun. **188**, 167 (2015).

[24] R. Prix and M. Shaltev, Search for continuous gravitational waves: Optimal StackSlide method at fixed computing cost, Phys. Rev. D **85**, 084010 (2012).

[25] K. Wette, Estimating the sensitivity of wide-parameter-space searches for gravitational-wave pulsars, Phys. Rev. D **85**, 042003 (2012).

[26] P. V. C. Hough, Machine analysis of bubble chamber pictures, in *2nd International Conference on High-Energy Accelerators and Instrumentation*, edited by L. Kowarski (CERN, Geneva, 1959), Vol. C590914, p. 554.

[27] B. F. Schutz and M. A. Papa, End-to-end algorithm for hierarchical area searches for long-duration GW sources for GEO 600, in *Gravitational Waves and Experimental Gravity*, edited by J. T. T. Vân, J. Dumarchez, S. Reynaud, C. Salomon, S. Thorsett, and J. Vinet (World Publishers, Hanoi, 1999), p. 199.

[28] B. Krishnan, A. M. Sintes, M. A. Papa, B. F. Schutz, S. Frasca, and C. Palomba, Hough transform search for continuous gravitational waves, Phys. Rev. D **70**, 082001 (2004).

[29] F. Antonucci, P. Astone, S. D'Antonio, S. Frasca, and C. Palomba, Detection of periodic gravitational wave sources by Hough transform in the f versus f-dot plane, Classical Quantum Gravity **25**, 184015 (2008).

[30] P. Astone, A. Colla, S. D'Antonio, S. Frasca, and C. Palomba, Method for all-sky searches of continuous gravitational wave signals using the frequency-Hough transform, Phys. Rev. D **90**, 042002 (2014).

[31] C. Palomba, P. Astone, and S. Frasca, Adaptive Hough transform for the search of periodic sources, Classical Quantum Gravity **22**, S1255 (2005).

[32] B. Krishnan and A. M. Sintes, Hough search with improved sensitivity, LIGO Technical Report T070124-x0, 2007.

[33] G. Mendell and M. Landry, StackSlide and Hough Search SNR and Statistics, LIGO Technical Report T050003-x0, 2005.

[34] V. Dergachev, Description of PowerFlux 2 algorithms and implementation, LIGO Technical Report T1000272-v5, 2010.

[35] V. Dergachev, On blind searches for noise dominated signals: a loosely coherent approach, Classical Quantum Gravity **27**, 205017 (2010).

[36] V. Dergachev, Loosely coherent searches for sets of well-modeled signals, Phys. Rev. D **85**, 062003 (2012).

[37] H. J. Pletsch and B. Allen, Exploiting Large-Scale Correlations to Detect Continuous Gravitational Waves, Phys. Rev. Lett. **103**, 181102 (2009).

[38] H. J. Pletsch, Parameter-space metric of semicoherent searches for continuous gravitational waves, Phys. Rev. D **82**, 042002 (2010).

[39] H. J. Pletsch, Parameter-space correlations of the optimal statistic for continuous gravitational-wave detection, Phys. Rev. D **78**, 102005 (2008).

[40] S. Walsh, M. Pitkin, M. Oliver, S. D'Antonio, V. Dergachev, A. Królak, P. Astone, M. Bejger, M. Di Giovanni, O. Dorosh, S. Frasca, P. Leaci, S. Mastrogiovanni, A. Miller, C. Palomba, M. A. Papa, O. J. Piccinni, K. Riles, O. Sauter, and A. M. Sintes, Comparison of methods for the detection of gravitational waves from unknown neutron stars, Phys. Rev. D **94**, 124010 (2016).

[41] R. Balasubramanian, B. S. Sathyaprakash, and S. V. Dhurandhar, Gravitational waves from coalescing binaries: Detection strategies and Monte Carlo estimation of parameters, Phys. Rev. D **53**, 3033 (1996).

[42] B. J. Owen, Search templates for gravitational waves from inspiraling binaries: Choice of template spacing, Phys. Rev. D **53**, 6749 (1996).

[43] R. Prix, Search for continuous gravitational waves: Metric of the multidetector F-statistic, Phys. Rev. D **75**, 023004 (2007).

[44] P. Astone, K. M. Borkowski, P. Jaranowski, and A. Królak, Data analysis of gravitational-wave signals from spinning neutron stars. IV. An all-sky search, Phys. Rev. D **65**, 042003 (2002).

[45] K. Wette and R. Prix, Flat parameter-space metric for all-sky searches for gravitational-wave pulsars, Phys. Rev. D **88**, 123005 (2013).

[46] K. Wette, Parameter-space metric for all-sky semicoherent searches for gravitational-wave pulsars, Phys. Rev. D **92**, 082003 (2015).

[47] K. Wette, Empirically extending the range of validity of parameter-space metrics for all-sky searches for gravitational-wave pulsars, Phys. Rev. D **94**, 122002 (2016).

[48] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Grundlehren der Mathematischen Wissenschaften Vol. 290 (Springer-Verlag, New York, 1988).

[49] B. J. Owen and B. S. Sathyaprakash, Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement, Phys. Rev. D **60**, 022002 (1999).

[50] P. Jaranowski and A. Królak, Gravitational-wave data analysis. Formalism and sample applications: The Gaussian case, Living Rev. Relativity **8**, 3 (2005).

[51] R. Prix, Template-based searches for gravitational waves: efficient lattice covering of flat parameter spaces, Classical Quantum Gravity **24**, S481 (2007).

[52] K. Wette, Lattice template placement for coherent all-sky searches for gravitational-wave pulsars, Phys. Rev. D **90**, 122010 (2014).

[53] C. Messenger, R. Prix, and M. A. Papa, Random template banks and relaxed lattice coverings, Phys. Rev. D **79**, 104017 (2009).

[54] G. M. Manca and M. Vallisneri, Cover art: Issues in the metric-guided and metric-less placement of random and stochastic template banks, Phys. Rev. D **81**, 024004 (2010).

[55] B. Knispel *et al.*, Einstein@Home discovery of 24 pulsars in the Parkes Multi-beam Pulsar Survey, Astrophys. J. **774**, 93 (2013).

[56] C. Messenger and A. Patruno, A semi-coherent search for weak pulsations in AQUILA X–1, Astrophys. J. **806**, 261 (2015).

[57] Einstein@Home, https://einsteinathome.org.

[58] P. Jaranowski and A. Królak, Data analysis of gravitational-wave signals from spinning neutron stars. II. Accuracy of estimation of parameters, Phys. Rev. D **59**, 063003 (1999).

[59] R. Prix and Y. Itoh, Global parameter-space correlations of coherent searches for continuous gravitational waves, Classical Quantum Gravity **22**, S1003 (2005).

[60] LALSuite, https://wiki.ligo.org/DASWG/LALSuite.

[61] D. C. Wells, E. W. Greisen, and R. H. Harten, FITS—A Flexible Image Transport System, Astron. Astrophys. Suppl. Ser. **44**, 363 (1981).

[62] D. Keitel, R. Prix, M. A. Papa, P. Leaci, and M. Siddiqi, Search for continuous gravitational waves: Improving robustness versus instrumental artifacts, Phys. Rev. D **89**, 064023 (2014).

[63] D. Keitel and R. Prix, Line-robust statistics for continuous gravitational waves: safety in the case of unequal detector sensitivities, Classical Quantum Gravity **32**, 035004 (2015).

[64] D. Keitel, Robust semicoherent searches for continuous gravitational waves with noise and signal models including hours to days long transients, Phys. Rev. D **93**, 084024 (2016).

[65] P. Morin, *Open Data Structures: An Introduction* (Athabasca University Press, Edmonton, 2013).

[66] J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring, *GNU Octave Version 4.0.0 Manual: A High-Level Interactive Language for Numerical Computations* (2015), https://octave.org/doc/octave-4.0.0.pdf.

[67] K. Wette, R. Prix, D. Keitel, M. Pitkin, C. Dreissigacker, J. T. Whelan, and P. Leaci, OctApps: A library of Octave functions for continuous gravitational-wave data analysis, J. Open Source Software **3**, 707 (2018).

[68] R. Prix, Characterizing timing and memory-requirements of the F-statistic implementations in LALSuite, LIGO Technical Report T1600531-v4, 2017.

[69] P. Patel, X. Siemens, R. Dupuis, and J. Betzwieser, Implementation of barycentric resampling for continuous wave searches in gravitational wave data, Phys. Rev. D **81**, 084032 (2010).

[70] P. R. Williams and B. F. Schutz, *An Efficient Matched Filtering Algorithm for the Detection of Continuous Gravitational Wave Signals*, edited by S. Meshkov, AIP Conference Series Vol. 523 (American Institute of Physics, Melville, 2000), p. 473.

[71] P. Leaci and R. Prix, Directed searches for continuous gravitational waves from binary systems: Parameter-space metrics and optimal Scorpius X-1 sensitivity, Phys. Rev. D **91**, 102003 (2015).

[72] S. Walsh, K. Wette, M. Papa, and R. Prix (to be published).