A binary tree approach to template placement for searches for gravitational waves from compact binary mergers

Chad Hanna,[1,2,3,4] James Kennington,[1,2,a] Shio Sakon,[1,2] Stephen Privitera,[5,b] Miguel Fernandez,[1,2] Jonathan Wang,[6] Cody Messick,[1] Alex Pace,[1,2] Kipp Cannon,[7] Prathamesh Joshi,[1,2] Rachael Huxford,[1,2] Sarah Caudill,[8] Chiwai Chan,[7] Bryce Cousins,[1,4] Jolien D. E. Creighton,[9] Becca Ewing,[1,2] Heather Fong,[7,10] Patrick Godwin,[1,2] Ryan Magee,[1,2] Duncan Meacher,[9] Soichiro Morisaki,[11] Debnandini Mukherjee,[1,2] Hiroaki Ohta,[7] Surabhi Sachdev,[1,2,12] Divya Singh,[1,2] Ron Tapia,[1,4] Leo Tsukada,[7,10] Daichi Tsuna,[7,10] Takuya Tsutsui,[7] Koh Ueno,[7] Aaron Viets,[9] Leslie Wade,[13] and Madeline Wade[13]

[1] *Department of Physics, The Pennsylvania State University, University Park, PA 16802, USA*
[2] *Institute for Gravitation and the Cosmos, The Pennsylvania State University, University Park, PA 16802, USA*
[3] *Department of Astronomy and Astrophysics, The Pennsylvania State University, University Park, PA 16802, USA*
[4] *Institute for Computational and Data Sciences, The Pennsylvania State University, University Park, PA 16802, USA*
[5] *Albert-Einstein-Institut, Max-Planck-Institut für Gravitationsphysik, D-14476 Potsdam-Golm, Germany*
[6] *Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, USA*
[7] *RESCEU, The University of Tokyo, Tokyo, 113-0033, Japan*
[8] *Nikhef, Science Park, 1098 XG Amsterdam, Netherlands*
[9] *Leonard E. Parker Center for Gravitation, Cosmology, and Astrophysics, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, USA*
[10] *Graduate School of Science, The University of Tokyo, Tokyo 113-0033, Japan*
[11] *Institute for Cosmic Ray Research, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8582, Japan*
[12] *LIGO Laboratory, California Institute of Technology, MS 100-36, Pasadena, California 91125, USA*
[13] *Department of Physics, Hayes Hall, Kenyon College, Gambier, Ohio 43022, USA*

(Dated: September 26, 2022)

arXiv:2209.11298v1 [gr-qc] 22 Sep 2022

We demonstrate a new geometric method for fast template placement for searches for gravitational waves from the inspiral, merger and ringdown of compact binaries. The method is based on a binary tree decomposition of the template bank parameter space into non-overlapping hypercubes. We use a numerical approximation of the signal overlap metric at the center of each hypercube to estimate the number of templates required to cover the hypercube and determine whether to further split the hypercube. As long as the expected number of templates in a given cube is greater than a given threshold, we split the cube along its longest edge according to the metric. When the expected number of templates in a given hypercube drops below this threshold, the splitting stops and a template is placed at the center of the hypercube. Using this method, we generate aligned-spin template banks covering the mass range suitable for a search of Advanced LIGO data. The aligned-spin bank required ~ 24 CPU-hours and produced 2 million templates. In general, we find that other methods, namely stochastic placement, produces a more strictly bounded loss in match between waveforms, with the same minimal match between waveforms requiring about twice as many templates with our proposed algorithm. Though we note that the average match is higher, which would lead to a higher detection efficiency. Our primary motivation is not to strictly minimize the number of templates with this algorithm, but rather to produce a bank with useful geometric properties in the physical parameter space coordinates. Such properties are useful for population modeling and parameter estimation.

I. INTRODUCTION

Banks of template gravitational-wave signals are central tools in the matched-filter detection of gravitational-wave signals from compact binary coalescence [1–3]. The general compact binary gravitational-wave signal depends on at least fifteen parameters: two mass parameters, six spin parameters, distance, time, and five angles defining binary orientation with respect to the gravitational-wave antenna. The parameter space can be even larger if, for instance, matter or eccentricity effects are included. Since we do not know the source parameters *a priori*, we must search the data over all possible source parameters.

We are often able to quickly maximize the signal-to-noise ratio (SNR) over a subset of the parameters either analytically or by efficient numerical techniques. For instance, some parameters[1] enter only into the overall amplitude of the signal which is normalized away by the

[a] james.kennington@ligo.org
[b] stephen.privitera@ligo.org

[1] Which parameters these are depends on the assumptions made about the signal. For instance, non-precessing binaries have a constant inclination angle, which enters into the gravitational-wave signal only in the overall scale of the waveform, whereas

matched-filter definition of SNR. The coalescence time enters into the waveform as a frequency-dependent phase shift which can efficiently be searched over using widely-available fast Fourier Transform routines. Considering only dominant $(\ell, |m|) = (2, 2)$ modes of gravitational-wave signals, the coalescence phase can also be maximized over analytically.

Given the approximations, assumptions and techniques described above, a subset of parameters, $\vec{\lambda}$, the *template bank* parameters, are generally relevant for template placement. We search over these parameters by laying down a discrete set of points in the parameter space and repeating the matched-filter calculation for each template. The set of points must be chosen as a compromise between optimal SNR recovery and available computational resources. Placing templates finely in the template parameter space leads to high SNR recovery, but can quickly make the search prohibitively expensive. In particular, the number of templates required to cover an D-dimensional parameter space such that no more than a fraction M of the SNR is lost to any potential signal scales as $M^{-D/2}$ [2].

In the case of non-spinning binaries, lattice placement strategies based on an approximate analytic expression for the signal space "distance" between two nearby templates have been shown to be effective for covering the template parameter space [4, 5]. To guarantee efficiency of the placement, these methods require that the metric $\mathbf{g}(\vec{\lambda})$, which defines the distance between nearby templates, is very nearly constant throughout the parameter space. For waveforms involving spin, in which a metric is either unavailable or varies rapidly throughout the parameter space, stochastic template placement has proven to be effective in covering the parameter space [6–10]. The stochastic placement technique works by randomly selecting a large number of points in parameter space and keeping only those points which fall sufficiently far away from points which have already been accepted into the bank. This technique, while robust, is computationally inefficient, although recent implementations have made significant strides towards optimization [9, 11, 12].

Geometric techniques have also been applied to generate aligned-spin template banks [13, 14]. In Ref. [13], the authors demonstrate a geometric template bank for neutron-star–black-hole binaries. The authors find satisfactory coverage for this parameter space by stacking two two-dimensional lattices, taking advantage of the fact that the parameter space is "thin" in the third dimension. This placement strategy was used in conjunction with ordinary stochastic placement [11] to cover the full compact binary parameter space searched in the recent LIGO-Virgo searches [15, 16]. In Ref. [14], the authors consider an interesting extension of this technique which starts with a true three-dimensional lattice, and falls back to the stochastic approach when the lattice approach breaks down. In Ref. [12], the authors also consider a hybrid stochastic-geometric technique, similar to the algorithm we propose here; however, the notion of lattice-adjacency the authors used is Cartesian whereas we incorporate the intrinsic geometry of the parameter manifold.

These solutions continue to rely at least partially on stochastic placement methods, which scales poorly with the number of templates. The required number of template parameters to cover a parameter space at a given minimal match threshold increases dramatically with the bandwidth of the interferometer and the dimension of the target signal space, both of which are ever-increasing in ground-based gravitational wave searches [11, 17, 18]. Currently used aligned-spin template banks have four template parameters (two masses and two spins) and over 1 million templates at maximal mismatches between 1–3% [19]. Precessional effects adds five more parameters (four spin components and the binary inclination at some reference frequency) and an additional order of magnitude in templates [18]. At high mass ratios, sub-dominant modes may also be important for detection, which can only further increase the template bank size. Presently template bank generation with stochastic methods may be computationally slow. Future larger banks will require more computing resources to generate as gravitational wave detector sensitivity improves. This can be problematic if banks are generated often.

Here, we demonstrate a new method for template placement based on a binary tree decomposition of the parameter space which is purely geometric originally explored here [20]. The algorithm relies on a numerical estimation of the parameter space metric and uses this metric to determine how to grow the binary tree. This algorithm requires $\mathcal{O}(2^n D^2)$ overlap calculations, where n is the bifurcation number of the parameter space, i.e., how many times a characteristic cell is split, and dim is the dimension of the resulting template bank. We demonstrate this method by constructing a bank suitable for Advanced LIGO and Advanced Virgo data analysis.

II. MOTIVATION

Beyond general interest in pursuing novel template placement algorithms, our motivation for pursuing this work is three-fold based on experiences analyzing LIGO and Virgo data during the third observing run. First, in order to apply a population model to gravitational wave detection, it is important to account for template placement [21] in a way that may account for the coordinate volume that a template occupies [22–25]. The binary tree approach that we have taken guarantees that each template ends up in a hyperrectangle in the physical coordinates making coordinate volume calculations easy. Second, in order to ensure a high availability of service for online compact binary searches we run searches at two

---

precessing binaries have a time-dependent inclination, leading to modulation in the waveform phase and amplitude.

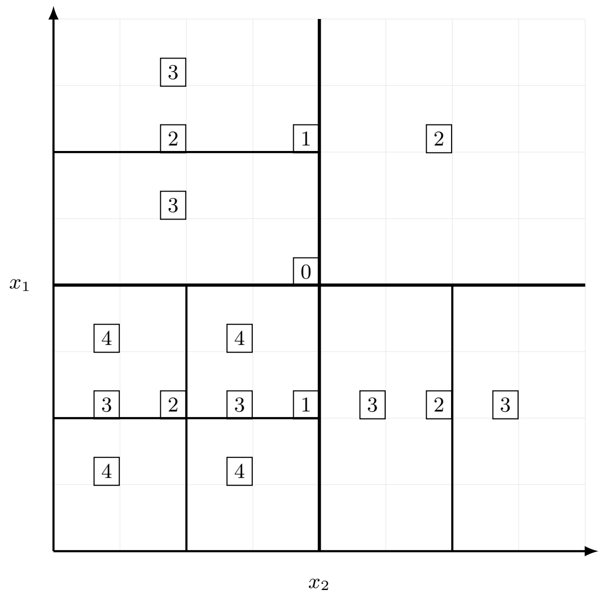

FIG. 1. Example hyper-rectangle bifurcation in two arbitrary dimensions, x_1, x_2. Each number, $\boxed{n}$, represents a place where the metric, $\mathbf{g}$ was computed at the *nth* stage of the bifurcation. This example results in nine hyper-rectangles, which is less than the maximum value of 2^4 after four bifurcations.

different data centers. The goal is to split the parameter space in a way that if one site goes down the other is still efficient at detecting a broad class of binary signals. The binary tree approach allows us to use a bank derived from the "right" and "left" splits separately. Finally, having a bank that is grid-like in physical coordinates is generally useful for template interpolation [26] and rapid parameter estimation [27] problems and we are interested in exploring this as future work.

III. METHODS

Our method, whose implementation we refer to as `treebank`, relies on having an accurate approximation of the template space metric $\mathbf{g}(\vec{\lambda})$, which gives a measure of the "distance" between nearby templates. For our work $\vec{\lambda} \equiv \{t_c, \log m_1, \log m_2, \chi_{\text{eff}}\}$, where $\chi_{\text{eff}} \equiv (m_1 a_{1z} + m_2 a_{2z}) / (m_1 + m_2)$ and a is the dimensionless spin [28]. We define the mismatch δ^2 between two nearby gravitational-wave templates, $h(\vec{\lambda})$ and $h(\vec{\lambda}+\vec{\Delta\lambda})$, according to

$$\delta(\vec{\lambda}, \vec{\Delta\lambda})^2 = 1 - \langle\, \hat{h}(\vec{\lambda}) \,|\, \hat{h}(\vec{\lambda} + \vec{\Delta\lambda})\, \rangle, \tag{1}$$

$$\langle a|b\rangle \equiv \left| \int_{-f_N}^{f_N} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)} df \right|, \tag{2}$$

where the template a or b is taken to be complex valued containing both the sine and cosine phases, thereby maximizing over phase, and f_N is the Nyquist frequency. δ^2 can be expressed in terms of a metric tensor $\mathbf{g}$ on the template signal manifold as

$$\delta(\vec{\lambda}, \vec{\Delta\lambda})^2 = \vec{\Delta\lambda}^T\, \mathbf{g}(\vec{\lambda})\, \vec{\Delta\lambda}. \tag{3}$$

From the metric, we can also compute a local volume element and thereby estimate the number of templates required to fill a given hypercube cell in the binary tree decomposition [2]:

$$\mathcal{N}_C(\vec{\lambda}) = \frac{\int \sqrt{|\det g(\vec{\lambda})|} dV}{V_T}, \tag{4}$$

where V_T is the volume of a template in mismatch space. We use the definition by Owen for the metric components in terms of the mismatch. [2]

$$g_{ij} = -\frac{1}{2} \left[\frac{\partial^2 \delta^2(\vec{\lambda}, \Delta\vec{\lambda})}{\partial \Delta\lambda^i \partial \Delta\lambda^j} \right]_{\Delta\lambda^k=0} \tag{5}$$

We have implemented two numerical schemes for estimating the metric component values that we call the iterative and deterministic methods. The iterative method is a standard convergence scheme for numerical differentiation leveraging the Python package numdifftools. The deterministic method uses definitions of the metric components as partial derivatives of the mismatch to compute the preliminary metric $\gamma_{\mu\nu}$ in a single step.

$$\begin{aligned} \gamma_{\mu\mu} &= \frac{\delta^2(\vec{\lambda}, \Delta\vec{\lambda})}{\Delta\lambda^{\mu 2}} \\ \gamma_{\mu\nu} &= \frac{\delta^2(\vec{\lambda}, \Delta\vec{\lambda}) - \gamma_{\mu\mu}\Delta\lambda^{\mu 2} - \gamma_{\nu\nu}\Delta\lambda^{\nu 2}}{2\ \Delta\lambda^\mu\, \Delta\lambda^\nu} \end{aligned} \tag{6}$$

Once the preliminary metric has been estimated using either method, we post-process the metric in two steps. First, we minimize $\gamma_{\mu\nu}\Delta\lambda^\mu\Delta\lambda^\nu$ with respect to the time lag between signals $\Delta\lambda^0$ by projecting out the time component of the metric estimate. This results in the adjusted, spatial metric components

$$g_{ij} = \gamma_{ij} - \frac{\gamma_{0i}\gamma_{0j}}{\gamma_{00}}. \tag{7}$$

Where we use the term *spatial* above to mean non-temporal, as in the familiar 3 + 1 decomposition. Second, we use an eigenvalue decomposition to check for numerical stability and validity of the estimated metric. If a negative eigenvalue is found, which would incorrectly imply a negative spatial signature, we attempt a reevaluation of the metric with a coarser set of intrinsic parameters $\vec{\lambda}' = \text{Coarse}(\vec{\lambda})$.

The template-bank algorithm then works as follows:

1. Initialize a hyper-rectangle bounding the parameter space one wishes to cover, e.g., a bounding box in component masses.

2. Compute the metric $\mathbf{g}(\vec{\lambda})$ numerically at the center of the hyper-rectangle. Alternatively, skip this step if the metric is sufficiently constant. We determine this by defining $\epsilon \equiv \left|1 - \sqrt{|g|}_{i-2} \;/\; \sqrt{|g|}_{i-1}\right|$ and setting a threshold on epsilon. In other words, if the volume element of the previous two iterations $(i-2, i-1)$ is sufficiently unchanged, the user may decide to skip this step. Setting epsilon to 0 forces the metric to be recomputed.

3. From the metric, estimate the number of templates $\mathcal{N}_C$ needed to cover this hyper-rectangle via Eq. 4.

4. If $\mathcal{N}_C$ is greater than the user-supplied threshold $\mathcal{N}_C^*$, compute the side lengths of the hyper-rectangle according to the metric and split the cube along its largest side in two children cells A and B. Call the algorithm recursively on A and B.

5. If $\mathcal{N}_C < \mathcal{N}_C^*$, place a template at the center of the cell and stop splitting. [2]

The splitting stops when all rectangles have $\mathcal{N}_C < \mathcal{N}_C^*$ or alternatively if the user specifies a minimum coordinate volume. In Fig. 1, we illustrate the decomposition.

Other than waveform generation, the most computationally costly step of this process is the evaluation of the mismatch between two templates (2), which is needed to evaluate the metric coefficients (5). In the case where the template parameter space is bifurcated n times, there will be at most 2^n hyper-rectangles. If $\epsilon = 0$, then the metric will be evaluated for every cell and,

$$\text{number of metric evaluations} = \sum_{i=0}^{n} 2^i = 2^{n+1} - 1. \quad (8)$$

Each metric evaluation requires $\mathcal{O}(D(D+1)/2)$ match calculations, where D is the dimension of the template parameter space, and the exact scaling depends on the finite differencing scheme chosen. This means that the total number of match calculations for a given bank assuming $\epsilon = 0$ is

$$\begin{aligned}\text{number of match evaluations} &= \frac{(2^{n+1}-1)D(D+1)}{2} \\ &= \mathcal{O}(2^n D^2)\end{aligned} \quad (9)$$

Each hyper-rectangle will contain one template, which means that a well balanced tree will contain a bank of $\mathcal{N}_B = 2^n$ templates. Thus, the number of match calculations *per waveform* in the template bank is

$$\frac{\text{number of match evaluations}}{\text{number of templates } (\mathcal{N}_B)} = \mathcal{O}(D^2). \quad (10)$$

The above gives a worst case scenario. Under normal circumstances $\epsilon > 0$ and the metric is found to be sufficiently constant that it does not need to be evaluated at the final tree depth. This leads to typical scaling where there are *fewer match calculations than there are templates in the bank $\mathcal{N}_B$*

By definitions the matches between waveforms used in the metric calculation are extremely high – approaching 1 minus floating point epsilon. Therefore, the function of frequency is extremely smooth and we evaluate waveforms and matches with extremely coarse spacing, typicallly 1 Hz.

IV. RESULTS

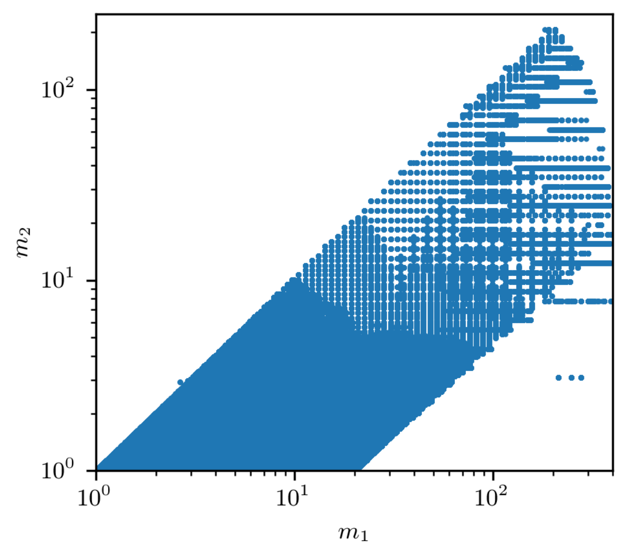

FIG. 2. Example template bank. This is a projection of the three dimensional bank in coordinates $\{\log m_1, \log m_2, \chi_{\rm eff}\}$ into the $\{\log m_1, \log m_2\}$ plane. The templates that appear to be outside of the region of interest have hyperrectangles that overlap with the region. Note that the naive template density is directly related to local volume element magnitude, and varies accordingly.

We used the algorithm described in the previous section to generate an advanced LIGO template bank using projected O4 sensitivity estimates [3]. We used a chirp

[2] Note that a single template is added to the bank even though N_C is an estimate of the number of templates to cover a hyper-rectangle and N_C^* can be greater than 1. In such a case, N_C^* acts as a coarse-graining parameter. We usually set $N_C^* \leq 1$.

[3] https://dcc.ligo.org/LIGO-T2200043

mass range from 0.87 – 174 $M_\odot$, a minimum secondary mass of 0.98 $M_\odot$, a maximum mass ratio of 20 and a maximum total mass of 400 $M_\odot$. We specified an effective spin range, χ, from -0.99 to 0.99 but limited the spin of objects below 3 $M_\odot$ to be less than 0.05. We allowed the template low frequency to go down to 10 Hz, but specified a maximum duration of 128s. We requested a maximum mismatch of 3%, but also set the minimum coordinate volume ($\Delta \log m_1 \times \Delta \log m_2 \times \Delta\chi$) to be greater than 0.0001. This resulted in 2,083,547 templates as shown in Fig. 2.

We validated the template bank by injecting 16,000 simulated signals in the parameter space. We find that the bank achieves the requested 97% match better than 99% of the time.

V. CONCLUSION

We have described here a new method for fast template bank placement, and shown that the method works in 3 dimensions relevant to dominant-mode aligned-spin template searches. The `treebank` method is computationally efficient and we expect this method will scale to higher dimensional template placement, such as precessing or sub-dominant mode templates, but we leave this for future work. It should also have applications in producing high density banks for use in rapid parameter estimation [27].

A tarball containing the source code necessary to reproduce the results in this paper can be found at https://pypi.org/project/gwsci-manifold.

ACKNOWLEDGEMENTS

This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center (https://www.gw-openscience.org/), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain.

This work was supported by National Science Foundation awards OAC-1841480, PHY-2011865, and OAC-2103662. Computations for this research were performed on the Pennsylvania State University's Institute for Computational and Data Sciences gravitational-wave cluster. CH Acknowledges generous support from the Eberly College of Science, the Department of Physics, the Institute for Gravitation and the Cosmos, the Institute for Computational and Data Sciences, and the Freed Early Career Professorship.

[1] B. S. Sathyaprakash and S. V. Dhurandhar, Phys. Rev. **D44**, 3819 (1991).
[2] B. J. Owen, Phys. Rev. **D53**, 6749 (1996), arXiv:gr-qc/9511032 [gr-qc].
[3] B. J. Owen and B. S. Sathyaprakash, Phys. Rev. **D60**, 022002 (1999), arXiv:gr-qc/9808076 [gr-qc].
[4] T. Cokelaer, Phys. Rev. D **76**, 102004 (2007), arXiv:0706.4437 [gr-qc].
[5] B. Abbott *et al.* (LIGO Scientific), Phys. Rev. **D78**, 042002 (2008), arXiv:0712.2050 [gr-qc].
[6] I. W. Harry, B. Allen, and B. Sathyaprakash, Phys. Rev. D **80**, 104014 (2009), arXiv:0908.2090 [gr-qc].
[7] S. Babak, Class.Quant.Grav. **25**, 195011 (2008), arXiv:0801.4070 [gr-qc].
[8] G. M. Manca and M. Vallisneri, Phys.Rev. **D81**, 024004 (2010), arXiv:0909.0563 [gr-qc].
[9] P. Ajith, N. Fotopoulos, S. Privitera, A. Neunzert, and A. J. Weinstein, Phys. Rev. D **89**, 084041 (2014), arXiv:1210.6666 [gr-qc].
[10] S. Privitera, S. R. P. Mohapatra, P. Ajith, K. Cannon, N. Fotopoulos, M. A. Frei, C. Hanna, A. J. Weinstein, and J. T. Whelan, Phys. Rev. **D89**, 024003 (2014), arXiv:1310.5633 [gr-qc].
[11] C. Capano, I. Harry, S. Privitera, and A. Buonanno, (2016), arXiv:1602.03509 [gr-qc].
[12] H. Fehrmann and H. J. Pletsch, Phys. Rev. **D90**, 124049 (2014), arXiv:1411.3899 [astro-ph.IM].
[13] I. W. Harry, A. H. Nitz, D. A. Brown, A. P. Lundgren, E. Ochsner, and D. Keppel, Phys. Rev. **D89**, 024010 (2014), arXiv:1307.3562 [gr-qc].
[14] S. Roy, A. S. Sengupta, and N. Thakor, (2017), arXiv:1702.06771 [gr-qc].
[15] B. P. Abbott *et al.* (Virgo, LIGO Scientific), Phys. Rev. **X6**, 041015 (2016), arXiv:1606.04856 [gr-qc].
[16] B. P. Abbott *et al.* (Virgo, LIGO Scientific), Astrophys. J. **832**, L21 (2016), arXiv:1607.07456 [astro-ph.HE].

[17] G. M. Harry and the LIGO Scientific Collaboration, Classical and Quantum Gravity **27**, 084006 (2010).

[18] I. Harry, S. Privitera, A. Bohé, and A. Buonanno, (2016), arXiv:1603.02444 [gr-qc].

[19] D. Mukherjee *et al.*, Phys. Rev. D **103**, 084047 (2021), arXiv:1812.05121 [astro-ph.IM].

[20] J. Wang, "Treebank: Differential geometric methods for fast template bank generation in searches for gravitational waves," (2017).

[21] T. Dent and J. Veitch, Phys. Rev. D **89**, 062002 (2014), arXiv:1311.7174 [gr-qc].

[22] H. K. Y. Fong, *From simulations to signals: Analyzing gravitational waves from compact binary coalescences*, Ph.D. thesis, University of Toronto (Canada) (2018).

[23] R. Magee *et al.*, Astrophys. J. Lett. **878**, L17 (2019), arXiv:1901.09884 [gr-qc].

[24] R. Abbott *et al.* (LIGO Scientific, Virgo), Phys. Rev. X **11**, 021053 (2021), arXiv:2010.14527 [gr-qc].

[25] R. Abbott *et al.* (LIGO Scientific, VIRGO, KAGRA), (2021), arXiv:2111.03606 [gr-qc].

[26] K. Cannon, C. Hanna, and D. Keppel, Phys. Rev. D **85**, 081504 (2012), arXiv:1108.5618 [gr-qc].

[27] C. Pankow, P. Brady, E. Ochsner, and R. O'Shaughnessy, Phys. Rev. D **92**, 023002 (2015).

[28] P. Ajith, M. Hannam, S. Husa, Y. Chen, B. Brügmann, N. Dorband, D. Müller, F. Ohme, D. Pollney, C. Reisswig, L. Santamaría, and J. Seiler, Physical Review Letters **106**, 241101 (2011).

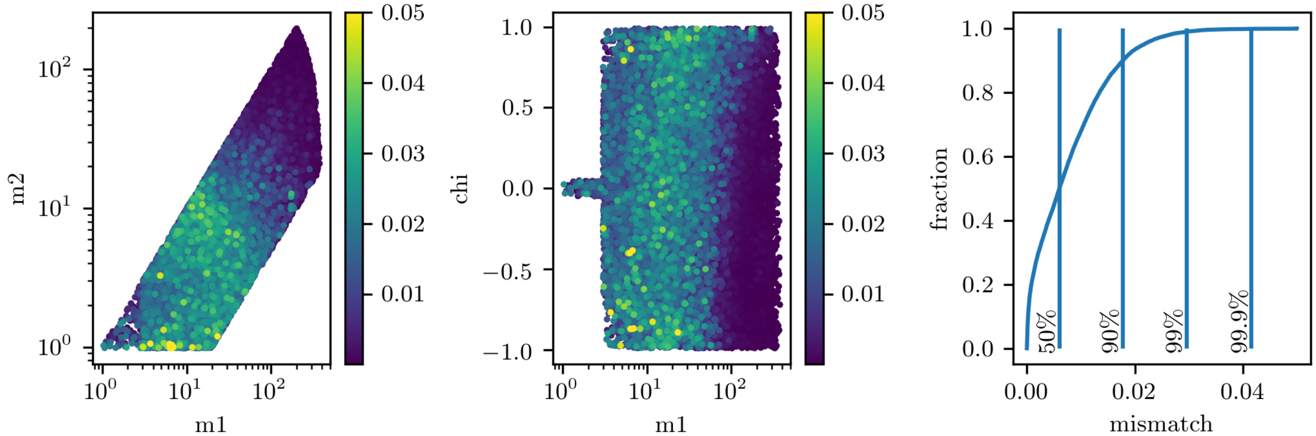

FIG. 3. Template bank validation. The bank achieves the requested 97% match 99% of the time and a better than 98% match 90% of the time. The large sample evaluation method used here is likely to be conservative since it does not check the match of all templates in the bank. The true performance may be better than this. The color bar indicates mismatch of simulated signal and nearest template. The injected signals were created using uniform distributions of the individual parameters $\{\log m_1, \log m_2 \chi_{\rm eff}\}$. The bank sim maximizes match only over nearby templates because the maximum match cannot decrease by including more templates. This balances computational speed for accuracy, but preserves acceptance criteria.