Hierarchical approach to matched filtering using a reduced basis

Rahul Dhurkunde, Henning Fehrmann, and Alexander H. Nitz
Max-Planck-Institut fur Gravitationsphysik (Albert-Einstein-Institut), D-30167 Hannover, Germany and Leibniz Universitat Hannover, D-30167 Hannover, Germany

(Received 9 November 2021; accepted 7 April 2022; published 2 May 2022)

Searching for gravitational waves from compact binary coalescence (CBC) is performed by matched filtering the observed strain data from gravitational-wave observatories against a discrete set of waveform templates designed to accurately approximate the expected gravitational-wave signal, and are chosen to efficiently cover a target search region. The computational cost of matched filtering scales with both the number of templates required to cover a parameter space and the in-band duration of the waveform. Both of these factors increase in difficulty as the current observatories improve in sensitivity, especially at low frequencies, and may pose challenges for third-generation observatories. Reducing the cost of matched filtering would make searches of future detector's data more tractable. In addition, it would be easier to conduct searches that incorporate the effects of eccentricity, precession or target light sources (e.g., subsolar). We present a hierarchical scheme based on a reduced basis method to decrease the computational cost of conducting a matched-filter based search. Compared to the current methods, we estimate without any loss in sensitivity, a speedup by a factor of $\sim$10 for sources with signal-to-noise ratio (SNR) of at least $= 6.0$, and a factor of $\sim$6 for SNR of at least 5. Our method is dominated by linear operations which are highly parallelizable. Therefore, we implement our algorithm using graphical processing units (GPUs) and evaluate commercially motivated metrics to demonstrate the efficiency of GPUs in CBC searches. Our scheme can be extended to generic CBC searches and allows for efficient matched filtering using GPUs.

DOI: 10.1103/PhysRevD.105.103001

I. INTRODUCTION

The first gravitational-wave (GW) detection in 2015 marked the dawn of GW astronomy [1]. The first two observation runs of LIGO [2] and VIRGO [3] detectors (O1 and O2) reported over a dozen confident detections [4,5]. The number of detections has rapidly increased to over 50 with the most recent O3 observing run [6,7]. To date, all gravitational-wave observations have come from compact binary coalescences (CBC); the vast majority of sources were from binary black holes (BBH) [6,7], but notably two binary neutron star (NS) mergers [8,9], and recently two neutron star—black hole NSBH mergers [10] have been observed. These observations have helped us to understand the physics of compact objects [11,12] and their dynamical evolution [11]. As the gravitational-wave observatories become more sensitive, the increased number of CBC sources will allow us to determine merger rate [13] and population distribution [14]. Upcoming third-generation observatories such as the Einstein telescope [15], cosmic explorer [16], and LISA [17] are expected to detect new kinds of astrophysical sources [15,18–20].

Matched filtering is the most widely used technique to detect CBC signals [5,21–23]. The method is optimal for stationary Gaussian noise [24]. While the detector data contains non-Gaussian noise transients [25,26], which require the use of vetoing techniques [27,28], matched filtering remains the dominant computational cost of a search algorithm [29]. In this work, we focus only on the implementation of matched filtering. Matched filtering requires accurate models of the expected gravitational waveform; CBCs can be modeled using different techniques [30–32]. The parameters of a binary merger are categorized into intrinsic (e.g., masses and spins) and extrinsic (e.g., binary orientation and location). To search for sources with unknown intrinsic parameters, we must select a discrete bank of templates which span the parameter space. These templates are chosen such that the minimum match (MM) between the data and at least one template from the bank is sufficiently large [33,34]. For example, a minimum match value of $\sim$0.97 would imply that at least 97% of the signal-to-noise ratio (SNR) of any signal with parameters within the search area could be recovered. To identify a potential signal, gravitational-wave strain data is convolved with every template in a bank to

calculate a signal-to-noise time series. Extrinsic parameters are often analytically maximized over. A possible candidate is identified if the SNR rises above a predetermined threshold, passes various tests of signal consistency [27,28] and data quality [25,35,36], and is statistically significant [4,37]. The computational cost of matched filtering, and so also the entire search, scales linearly with the number of templates in a bank and also increases with the duration of the observable signal, though generally sublinearly.

With advancements in the current and future detectors, it is expected that observation of signals at an increasingly lower frequency will become possible [15,18,19,38,39]. As the low frequency cutoff of a search decreases, both the size of the template bank and the signal duration grows rapidly, leading to increased computational costs [38,40,41]. While current template-based searches have confident detections only from quasicircular aligned-spin binaries [5,6,21,22], sources that exhibit measurable eccentricity or precession of the orbital plane could provide unique astrophysical insights [42,43]. While a few searches have included the effect of eccentricity [40,44,45] for parts of parameter space, many searches neglect the effects from eccentricity of the orbital plane [46] and precession of the orbit [47] in part due to the increased computational cost relative to normal searches [7,40,41,48]. For example, it has been shown that the template bank including precession is at least 10 times bigger than one without precession [48]. Furthermore, the lower mass boundary of subsolar primordial black hole searches is limited by computational cost considerations [7,38,40,41]. Development of a cost-efficient filtering algorithm would allow searches to be conducted more easily, with higher sensitivity, and in uncharted regions of parameter space.

The computational costs for matched filtering data with templates consist of redundant computations due to significant overlap of templates with each other in the neighborhood. This redundancy is be eliminated by using an orthonormal basis to filter data instead of templates [49]. The costs of filtering scale linearly with the number of basis and can be reduced by rejecting the basis vectors of lower importance [49]. Disregarding contributions from a few basis vectors leads to a loss in SNR, but this loss is kept under a tolerance by tuning the number of relevant basis p involved for filtering purpose [49]. When considering large number of templates, the value of p is much smaller, and therefore, it is possible to filter data with a fewer number of basis. Current online (low-latency) searches [23,50] employ this technique and are in good agreement with searches that do not use this approximation [51,52].

Since the basis vectors do not correspond to any physical source, matched-filtering outputs from each basis are weighted and linearly combined to give an SNR time series for a unique template, in a *reconstruction* process. The reconstruction is performed for each template and at every time sample which incurs additional costs to matched filtering. The naive costs for reconstructing SNR time series for T templates can be estimated in terms of a matrix multiplication which scales as $\mathcal{O}(NpT)$, where p is the number of basis vectors and N is the number of samples in the data. On the other hand, the direct template-based filtering is widely done using a Fast Fourier Transform (FFT)—based algorithm which requires $\mathcal{O}(NT\log N)$ operations. Usually $\log N \sim O(10)$ and $p \sim O(10^2$–$10^3)$ which suggests that naive reduced basis filtering is more expensive than the template-based filtering.

In this work, we demonstrate a cost-efficient matched filtering method by employing a new hierarchical scheme using a reduced set of basis. The reduced basis are obtained by applying principal component analysis (PCA) on a template bank. A two-stage hierarchical scheme is then invoked to compute the SNR time series for each template. In the first stage, an intermediate time series is computed that corresponds to a binned average of the complete SNR time series. In the second step we, do a full time resolution (nonaveraged) reconstruction using the reduced basis inside the bins where the average SNR exceeds a threshold. We demonstrate our method on simulated Gaussian noise and a population of CBC signals. To estimate the improvement from our method, we compare it against the flat template-based filtering scheme used in current searches. We observe that our method attains a speed up by a factor of $\sim$6 for a threshold of SNR $= 5$. Furthermore, we expect the performance of our method to increase at higher SNR thresholds, and similarly observe a performance gain of $\sim$10 times for SNR $= 6$.

Matched filtering is dominated by mathematical operations which are easily parallelizable across different threads or computation cores. Since a GPU is designed to accommodate a large number of threads, we employ GPUs in this work for an efficient implementation of matched filtering in parallel. To investigate the relative performance of different hardware, we compare the matched filtering implementations on GPUs with the central processing units (CPUs) that is currently used in the PyCBC search pipelines. We use two metrics to quantify the performance—cost, and power efficiency—while filtering data. We observe that GPUs are much more efficient than CPUs in performing matched filtering. In this analysis, we have restricted ourselves to aligned spins, dominant mode, and a single-detector analysis. However, our method can be easily extended to multiple detectors, and search scenarios including eccentricity, precession, or higher-order modes.

The rest of the paper is organized as follows. In Sec. II, we give a brief overview of the matched filtering process and the motivatation for why we need a new approach which is different from previous works. In Sec. III we describe our search method and the implementation. In Sec. IV we presents our results and compare them with the existing search methods. Finally we conclude in Sec. V.

II. COMPACT BINARY COALESCING SEARCHES

The core of any modeled searches for CBC signals is the matched-filtering technique which involves searching the interferometric data for a modeled waveform of the expected GW signal [5,21–23]. In this section we briefly describe the process of matched filtering and introduce some ideas for efficient filtering algorithms. We also discuss previous efforts to improve the performance of matched filtering in Sec. II B.

GW signals from a noneccentric CBC sources are characterized by 15 parameters [30–32]. These parameters are divided in two categories, (1) intrinsic parameters—(m_1, m_2) component masses, and three-dimensional spin vectors ($\vec{\chi_1}$,$\vec{\chi_2}$) and (2) extrinsic parameters (in the observer frame of reference)—standard spherical coordinates (D, i, ψ), sky-location (θ, ϕ), and lastly (t_c, ϕ_c) the time and the phase at the coalescence. The anticipated signal is then accurately modeled in terms of these parameters with the help of various analytical and numerical techniques [30–32].

To search for the modeled signal $\tilde{h}(f)$ also referred as template, matched filtering is performed in the Fourier domain to quantify the likelihood of data containing the particular template. The matched-filter statistic is a correlation between the Fourier transform of the data $[\tilde{s}(f)]$ and the template $[\tilde{h}(f)]$ weighted by the noise power spectral density (PSD) $S_n(f)$ [24]. It can be shown that the matched filter is an optimal detection statistic for distinguishing signals in the presence of stationary Gaussian noise [24]. The mathematical form of the complex matched filter statistic is

$$\langle s|h\rangle = 4\int_0^\infty \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_n(f)}df. \tag{1}$$

The output of the matched filter after normalizing with the correlation of the template with itself $\langle h|h\rangle^{1/2}$ is the signal-to-noise ratio

$$\rho^2 = \frac{(\mathrm{Re}[\langle s|h\rangle])^2}{\langle h|h\rangle}. \tag{2}$$

A priori the parameters of gravitational-wave signals are unknown and to search for the intrinsic parameters, a discrete template bank is used to *cover* the intrinsic parameter space. The notion of cover is to sample enough points in the parameter space such that the match between data and at least one template is above a minimum match value. In current searches typically a minimum match of 0.97 is used [29,53], and lattice-based [53–55], stochastic methods [56,57] or hybrid methods [33,34] are applied to sample the points in the parameter space. Since we are considering aligned spins with the orbital angular momentum in the $+z$ direction, the intrinsic parameter space consists only of $\zeta = (m_1, m_2, \chi_{1z}, \chi_{2z})$ parameters. The search over the two categories of binary parameters are handled differently—the intrinsic parameters are searched for by repeatedly match filtering for every template, whereas, the extrinsic parameters—sky location, the orientation of the binary, and distance to the source are accounted for as an overall phase ϕ_0 and an amplitude A [29]

$$\tilde{h} = Ae^{i\phi_0}\tilde{h}_0(\zeta)e^{2\pi i f t_c}. \tag{3}$$

In Eq. (3), A and ϕ_0 are unknown functions of $(D, i, \theta, \phi, \psi)$, and h_0 depends on the intrinsic parameters. The unknown amplitude A is a nuisance parameter that is eliminated by normalizing the SNR with the norm of the template as seen in Eq. (2). The unknown phase ϕ_0 is maximized using a quadrature, which is equivalent to maximizing the norm of the complex SNR [29]. Finally, the position of the signal is determined by searching for the time of coalescence of binary, represented by the t_c parameter. Variation in t_c is expressed as time-translations, and is separated using $e^{2\pi i f t_0}$. Substituting Eq. (3) into Eq. (1), the matched-filter output (SNR) at $t = t_0$ is given by Eq. (4). The SNR as a function of time can be obtained efficiently by performing an inverse FFT (IFFT) of Eq. (4) [29,54]

$$\langle s|h_0\rangle(t_0) = 4\int_0^\infty \frac{\tilde{s}(f)\tilde{h}_0^*(f;\zeta)}{S_n(f)}e^{2\pi i f t_0}df. \tag{4}$$

A data segment with a SNR above a predetermined threshold is referred to as a *trigger* which may contain a true GW signal. The ambiguity is due to the assumption of stationary Gaussian noise for the matched-filter statistic. However, triggers due to nonstationary glitches or pure Gaussian noise can give rise to false alarms which lower our confidence of identifying true GW signals [25,26]. The additional signal-consistency test introduced in [27,28,58] is performed to down-rank triggers due to glitches. Furthermore, it is ensured that only coincident triggers from multiple detectors are considered i.e., triggers corresponding to the same template and observed within the light travel time window between the detectors [59]. Amongst the various steps mentioned, matched filtering comprises the dominant computational costs of a search. Hence, our focus is to optimize the matched-filtering process.

To summarize the matched-filtering procedure, the intrinsic and extrinsic parameters are searched for separately using a template bank and analytical techniques, respectively. First, the PSD weighted correlation of data with a single template in ζ is computed, and then an inverse FFT is executed to obtain a complex time series. Taking the modulus of the complex times series and normalizing it by the norm of the template gives the SNR time series. The

above steps are repeated for all the templates in the template bank to search over the intrinsic parameters. Throughout this paper, we refer to the method of matched-filtering data with the templates as the *template method* for simplicity.

It is clear from above that matched-filtering operation scales linearly with the number of templates. In the case when the size of the template bank is large, a search can be limited by the computational costs required for filtering the data with templates. It is possible to numerically reduce the size of the template banks to a fewer number of basis vectors and filter data directly with the basis. We now give a brief introduction to performing matched filtering with a reduced basis.

A. Matched filtering using a reduced basis

Consider a region in the parameter space described by $\zeta = (m_1, m_2, \chi_{1z}, \chi_{2z})$. Discrete templates are used to cover this region, and as a result of the mismatch criterion, templates are strongly correlated in the vicinity of each other. The correlation between the templates incurs a redundancy in matched-filtering computations; instead, an orthonormal basis can be used to eliminate these correlated computations [49]. Commonly used methods for computing an orthonormal basis is the principal component analysis (PCA) and the singular value decomposition (SVD). The PCA approach to obtain the basis is found by performing an eigenvalue decomposition (EVD) of the covariance matrix $\mathbf{C} = \mathbf{T}^\top \mathbf{T}$ constructed using the templates [see Eq. (5a)], whereas, SVD is applied directly to a matrix containing the templates $\mathbf{T}$ [see Eq. (5b)].

$$\mathbf{C} = \mathbf{P}\mathbf{L}\mathbf{P}^\top, \tag{5a}$$

$$\mathbf{T} = \mathbf{U}\mathbf{S}\mathbf{P}^\top. \tag{5b}$$

In the case when templates are centered the column means of $\mathbf{T}$ are zero, then both SVD and PCA yield the same orthonormal basis. The basis is represented by the columns of $\mathbf{P}$ which are ranked by their corresponding eigenvalues in $\mathbf{L}$ or singular values in $\mathbf{S}$. It can be easily shown that $\mathbf{L} = \mathbf{S}^2$ and that the two methods are similar. Hence, either of the methods are applicable to obtain the basis.

Consider a set of p_t basis vectors for the parameter region ζ denoted by $\tilde{p}(f)$ in the Fourier domain. Every template in this region h_ζ is expressed in terms of a unique linear combination of the basis. Since the matched-filtering operation is also linear, we can filter the data $\tilde{s}$ only using the basis and rewrite Eq. (4) in terms of $\tilde{p}(f)$

$$\rho_r(t) = \sum_{k=0}^{p_t-1} 4 \int_0^\infty \frac{\tilde{s}(f) c^*_{k,\zeta} \tilde{p}^*_k(f)}{S_n(f)} e^{2\pi i f t} df, \tag{6}$$

where the template $\tilde{h}_\zeta(f) = \sum_{k=0}^{p_t-1} c_{k,\zeta} \tilde{p}_k$, is a linear combination of the basis $\tilde{p}(f)$ and the unique decomposition coefficients $c_{k,\zeta}$. The coefficient $c_{k,\zeta}$ is obtained by computing the scalar product of $\tilde{h}_\zeta$ and the kth basis vector $\tilde{p}_k$.

Matched filtering with the basis is done by first performing an IFFT of the correlation between $\tilde{s}$ and $\tilde{p}_k$, which results in a complex time series defined as $\beta_k(t) = \mathrm{IFFT}(\langle s | p_k \rangle)$. Afterwards, each β_k is weighted accordingly using the respective decomposition coefficients and then combined to give SNR time series corresponding to the template $\tilde{h}_\zeta(f)$. The process of multiplying the coefficients $c_{k,\zeta}$ with β_k is the reconstruction step, as it reconstructs the SNR time series using the contribution from the basis

$$\rho_r(t) = \sum_{k=0}^{p_t-1} c^*_{k,\zeta} \beta_k. \tag{7}$$

Using the complete orthonormal basis $p_t = T$ where T is the number of templates, Eq. (7) reproduces the same exact results as Eq. (4).

Instead of using the full basis, it is possible to approximately reconstruct the SNR with fewer basis vectors. Eigenvalues σ are arranged in decreasing order and only the first p basis vectors are chosen and the rest are discarded. It can be shown that the first p basis vectors span an approximate lower-rank subspace of the original parameter space. Neglecting contributions from some basis vectors leads to an average loss in SNR, which is shown to be a function proportional to the eigenvalues [49]

$$\left\langle \frac{\delta\rho}{\rho} \right\rangle = 1 - \frac{|\sum_{k=0}^{p-1} \sigma_k^2|}{|\sum_{k=0}^{p_t-1} \sigma_k^2|}. \tag{8}$$

The equation above indicates that the number of relevant basis p can be fine tuned based on the choice of tolerance in loss of SNR. For detection purposes, we want to keep this loss under the mismatch value (0.03) due to the discreteness of the template bank.

Reconstruction of the SNR time series is performed for every template and thus, the reduced basis approach requires additional costs to matched filtering. To estimate the reconstruction costs in brief (exact costs are estimated later in this paper), consider a data segment having N samples, T number of templates, and reduced p basis vectors. The number of operations required for the reconstruction step is $O(NpT)$. Meanwhile, comparing the costs to template-based filtering which requires $O(NT \log N)$, the actual comparison boils down to $\log N$ and p. The exact values for $\log N$ and p vary over the parameter region, but it is observed that p is typically 1–2 orders in magnitude bigger than $\log N$. Hence, the reduced-basis approach loses all the computational advantage of filtering against fewer basis.

B. Comparison with current methods

Different methods in the past have been implemented to speed up the process of filtering either by reducing the latency of the search [22,60,61] or by decreasing the required computational costs [62–64]. For the former case, a reduced-basis filtering technique along with multirate sampling is used with an intent to decrease the latency of matched filtering. In the latter case, the aim is to reduce the filtering costs by using a multistage hierarchical filtering method. Some of these techniques have been already implemented in the current search pipelines [23,50,52], and are deployed in different search scenarios. We briefly discuss these various strategies and contrast our methodology next.

1. Reduced basis filtering with or without multirate sampling

To discretize a continuous signal, the Nyquist-Shannon criterion [65] determines the sampling rate to be at least twice the highest resolvable frequency ($1/dt \geq 2f_{\max}$) of the anticipated signal. This gives the relation for the number of samples in the data $N = (dfdt)^{-1}$ where $1/dt, 1/df$ are sampling rate and sampling frequency respectively, suggesting the filtering costs increase when searching for higher frequencies. Because the frequency evolution of these signals is chirplike, rapidly increasing towards the merger, this allows a low sampling rate at the earlier times and can be increased subsequently as the signal evolves. Using multiple sampling rates the matched filtering costs are reduced significantly [22,60,61].

Multirate sampling has been adopted by the MBTA [22], LLOID [61], and SPIIR [60] schemes that are implemented in current online pipelines for the prompt detection of signals [22,23,50]. The MBTA method performs matched filtering in the Fourier domain using the standard FFT approach to obtain a SNR time series, whereas the LLOID and SPIIR methods perform time-domain filtering by employing FIR or IIR filters respectively to compute an equivalent form of the matched filter [Eq. (4)]. These filters are specially designed for whitening the data, a process which causes the most latency in matched filtering [22,23,50]. The overall latency is further improved by using a reduced basis obtained by performing SVD of the IIR/FIR filters. Results from the online pipelines are very well in agreement with the rigorous offline searches [21,51], justifying the viability of multirate sampling and reduced basis filtering in CBC searches.

The number of templates T drastically increases when searching in subsolar regions [7,38,40,41] or with additional parameters [40,48] e.g., eccentricity. In such search scenarios obtaining a reduced basis for the complete template bank is computationally limited. This is because SVD is performed on a template matrix whose size scales linearly with T and might require infeasible amounts of memory for the template matrix. To address this issue we choose the PCA approach of computing the basis, which is performed on a covariance matrix whose size is independent of T, and therefore, making it feasible to obtain orthonormal basis even for large template banks.

The major drawback of reduced-basis filtering is the large reconstruction cost, and hence, this method is avoided in extensive offline searches where the computational costs play an important role. In [66,67] the authors have introduced a new technique to decrease the reconstruction costs by using a random projections (RP) based reduced basis filtering method. Another approach to reduce the total costs is to split the matched filtering into multiple stages in a hierarchical fashion. Next, we discuss established hierarchical methods and compare them with our new hierarchical scheme.

2. Hierarchical methods

The crux of any hierarchical search is to perform a coarse and a fine search over the parameters involved in the hierarchy. In the past, the work in [63] proposed a two-stage hierarchical filtering on just a single chirp mass parameter, and the same work was extended in [64] for three parameters—the component masses and the time of coalescence. The most recent works [62,68] in the hierarchical approach to matched filtering extended the scheme to multiple detectors analysis. In the first stage, data is downsampled at 512 Hz, and filtered using a coarse bank of MM = 0.9. In the second stage, data is sampled at the full rate of 2048 Hz or 4096 Hz, and filtered with a fine bank having MM = 0.97. Their method achieved ∼20× speed up in comparison to the one-step search on simulated data containing only Gaussian noise. The latter work in [68] hierarchically searched advanced LIGO's first two observing runs and recovered all the events presented in the GWTC-1 catalog [4].

To assign significance of any event, it is important to estimate the noise background which is the trigger distribution due to noise only events [59]. In the work [62,68], the performance gain comes at the expense of a poor estimation of the background. Since they do not follow the noise triggers until the second stage, the true background is mimicked by scaling the first stage background. This leads to an improper estimation of the significance for an event.

In this method, we present a new hierarchical method aimed at reducing the reconstruction costs. We perform a two-stage hierarchical reconstruction of the SNR time series for the complete bank. We follow up all the triggers for SNR ≥ 5 till the second stage, and therefore, are able to accurately determine the original noise background. Furthermore, our method incurs no loss in the search sensitivity. We discuss the methodology in detail in the next section.

III. REDUCED BASIS HIERARCHICAL MATCHED FILTER

In this section we describe our new hierarchical approach to reduced basis matched filtering in detail. We first briefly review the PCA method in general and discuss a nonuniform sampling technique to reduce the costs of performing PCA. Then we explain how PCA is applied on a template bank and the hierarchical method of filtering data along with their implementation on GPUs. Finally, we estimate the matched filtering costs in detail for the template-based method and the reduced basis hierarchical scheme to compare the relative gain in performance.

A. PCA using nonuniform sampling

We first briefly explain the PCA procedure applied on n vectors denoted by $\mathbf{v}$. Every vector is centered by subtracting the mean vector $\mathbf{v}_s = \mathbf{v} - \mathbf{b}$, where $\mathbf{b} = 1/n \sum_{i=0}^{n} v_i$ is the mean vector. To ensure each vector gets equal weight, they are normalized with respect to the inner product defined on the vector space. Using the normalized and centered vectors $\hat{\mathbf{v}}_s$, a covariance matrix is created $\mathbf{C} = \hat{\mathbf{v}}_s^{\top} \hat{\mathbf{v}}_s$. An EVD is performed to get the orthonormal basis vectors $\mathbf{p}_t$ of the $\mathbf{C}$ matrix, which are ranked by the corresponding eigenvalues σ. The basis vectors corresponding to small eigenvalues are discarded, and the resulting set of reduced basis is denoted by $\mathbf{p}$. Projecting $\hat{\mathbf{v}}_s$ onto $\mathbf{p}$ gives the decomposition coefficients $\mathbf{D} = \mathbf{p}\hat{\mathbf{v}}_s$. The reduced basis $\hat{\mathbf{v}}_s$ and the decomposition coefficients $\mathbf{D}$ are used to retrieve an approximate version of $\hat{\mathbf{v}}_s$, given by $\hat{\mathbf{v}}_s^{\text{approx}} = \mathbf{D}^{\top}\mathbf{p}$.

Even though the costs of PCA are amortized, PCA can be time intensive and difficult to perform on a large collection of longer duration templates e.g., corresponding to lower masses, or with lower-frequency cutoffs. Amongst the various steps involved in PCA, the dominant computational and memory costs are for the covariance matrix—both scale quadratically with the number of samples N_t required for the templates. To put the scaling relations into perspective, consider the complete O2 bank sampled at a constant sampling frequency of 128s, the estimated memory required for $\mathbf{C}$ is ∼550 GBs. Distribution of the EVD process for $\mathbf{C}$ across several machines is a difficult task, and thus, the size of the covariance matrix is constrained due to the memory of a single machine. Hence, it is crucial to reduce the size of the templates to make the PCA faster and feasible in the low-mass regime.

In this work, we consider the frequency range from [15, 1024] Hz. The principle idea behind efficient sampling is to adjust the sampling rate according to the number of oscillations of a function within a given frequency bin. To account for the complete frequency range in our sampling analysis, we consider the template with the longest bandwidth and identify all the frequencies corresponding to the zero crossings of this template. The identified frequencies are used to define the edges of the nonoverlapping bins of different sizes. We further sample every bin by using five uniformly-spaced frequencies within, and together they make up the complete set of nonuniform sampling frequencies. We also ensure that our sampling criteria is never less than $df = 1/128$, which helps us avoid oversampling the dense bins at very low frequencies. The number of frequencies per bin is chosen empirically based on the relative error induced in the templates for not using the full sampling rate. Using this scheme we obtain a much smaller set of frequencies; $N_t = 11074$ compared to $N = 2048 \times 128$ uniform frequencies for efficient sampling of the templates and the basis. Once the basis are obtained, we interpolate them back to the original sampling frequencies.

To test the accuracy of our sampling method, we check the overlap between the templates generated using nonuniform and uniformly sampled frequencies. For this purpose, templates evaluated at nonuniform frequencies are linearly interpolated to a constant sampling frequency of 128s. We obtain a mismatch of $<10^{-4}$ in the overlap due to the interpolation of the templates, which is much smaller than the error due to the discreteness of the template bank and, therefore, can be safely neglected. This justifies that our nonuniform frequencies are viable for sampling the templates. Using our sampling method significantly reduces the memory required for $\mathbf{C}$ from ∼550 GBs to only ∼0.9 GBs—therefore, saving a lot of computational resources and simultaneously speeding up the PCA process.

B. Implementing PCA on a template bank

In this work, we use the template bank as described in [69] which was also used for the PyCBC analysis of the O2 observing run [5]. The parameter ranges used in this bank are total mass $M \in [2, 100]$ and mass ratio $q \in [1, 98]$. We restrict ourselves to the aligned-spin case where the spins for NS are up to 0.05 and up to 0.998 for BHs. The minimum match criterion used in this bank is 0.97, and the bank contains $T_{\text{total}} \sim 400,000$ templates.

We divide the parameter space into smaller regions to reduce the number of local basis p as they contribute linearly to the dominant reconstruction costs. The complete bank is split into smaller sub-banks, and then PCA is performed on each of them individually. Splitting of template bank is performed in the (τ_0, τ_3) coordinates along iso-τ_0 lines because the metric is roughly Euclidean in these coordinates [54]. The parameter τ_0 roughly corresponds to the duration of a template (in seconds) that scales as $\tau_0 \propto \mathcal{M}^{-5/3} f_0^{-8/3}$, where $\mathcal{M}$ is the chirp mass and f_0 is the lower frequency used in the analysis. We choose to split the τ_0 range into 64 equal parts, each of them containing 6250 templates. While optimizing the splitting is not in the scope of this work, we performed empirical testing of the number of splits by considering smaller or

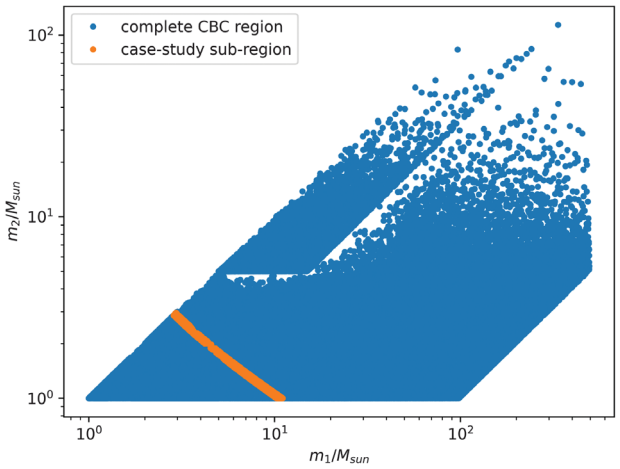

FIG. 1. O2 template bank [69] used in this work; the small orange region corresponds to the sub-bank used as a case study containing 6250 templates.

bigger equal parts than 64, and observed no significant improvement. In Fig. 1 we show the complete parameter space along with an example subregion which is used as a case study for further analysis.

The PCA operation begins by sampling templates at the previously obtained N_t nonuniform frequencies using the IMRPhenomPv2 waveform model [30]. Templates are then whitened using the aLIGO PSD and normalized to unity. The template matrix for the mth sub-bank $\mathbf{T}^m$ is constructed by storing templates row wise such that $\mathbf{T}^m$ has the dimensions of $(T \times N_t)$, where $T = T_{\text{total}}/64$. In the case when T_{total} is not a multiple of 64, we can simply choose another divisor close to 64. We observed the mean vector of $\mathbf{T}^m$ to be almost zero, and hence, skip the mean subtraction step. Covariance matrices for each sub-bank $\mathbf{C}^m$ are evaluated by multiplying the template matrix with its transpose $\mathbf{T}^m \times (\mathbf{T}^m)^\top$. In the next step, we perform the EVD of $\mathbf{C}^m$ to obtain the basis vectors and their corresponding eigenvalues. For this purpose we employ the Lanczos algorithm [70]—an efficient algorithm to obtain the p largest eigenvalues. Invoking Eq. (8) for a tolerance of 10^{-5}, the number of relevant eigenvalues p obtained for a few different sub-banks are shown in the Table I.

TABLE I. Example of a few sub-banks corresponding to their respective τ_0 ranges in the second column. The third column is the number of relevant eigenvalues p for the respective subregion.

Sub-bank index	τ_0 (sec)	p
1	[0.1, 5.1]	64
…	…	…
…	…	…
34(case study)	[98.0, 103.4]	254
…	…	…
…	…	…
64	[442.5, 595.7]	200

We then compute the decomposition coefficients essential to reconstruct the whitened templates in $\mathbf{T}^m$. These coefficients are unique for each template and are obtained by multiplying the two matrices—basis matrix $\mathbf{P}^m$ and the template matrix $\mathbf{T}^m$. The resulting matrix is the decomposition matrix $\mathbf{D}^m$ containing the p unique coefficients for every template in the mth sub-bank and has the dimensions $(p \times T)$. Finally, the original whitened templates can be approximately reconstructed by multiplying $\mathbf{D}$ and $\mathbf{P}$. In the next section, we discuss the two-stage hierarchical reconstruction of the matched filter output using $\mathbf{D}$ and $\mathbf{P}$.

C. Hierarchical reconstruction of the SNR time series

Matched filtering with the basis vectors as per Eq. (7) is performed using the following steps:

(i) compute FFT of the data $s(t)$ with N sample points at a uniform sampling rate $1/dt$ to obtain $\tilde{s}(f)$.
(ii) linearly interpolate the basis vectors $\tilde{p}_k(f)$ at the uniform frequencies (multiples of $2f_{\max}/N$).
(iii) filter data with every basis vector to obtain β_k—inverse FFT of the product $\tilde{s}(f)\tilde{p}_k^*(f)/\sqrt{S_n(f)}$.
(iv) average β_k in bins of w samples to obtain β_k^{avg}.
(v) perform first stage reconstruction to obtain averaged SNR time-series.
(vi) perform second stage reconstruction around the triggers from the first stage.

The first step in the reduced basis matched filtering process is to compute forward FFT of the data $s(t)$. Since $s(t)$ is sampled uniformly with a rate of $1/dt = Ndf$, the Fourier-transformed data $\tilde{s}(f)$ is obtained at frequencies given by integer multiples of $2f_{\max}/N$, where $f_{\max} = 1/(2dt)$. Now to filter the data with the basis, the correlation of data and basis are computed at the uniform frequencies, and for this reason, the basis are linearly interpolated from $f_{\min}$ to $f_{\max}$ in df steps. Since the basis are already whitened and the denominator in the matched filtering [Eq. (6)] requires $S_n(f)$, we multiply the correlation product with $S_n(f)^{-1/2}$ to get the appropriate denominator. For a basis vector $\tilde{p}_k$, the filtered output β_k time series is obtained by computing the inverse FFT of the weighted correlation. Finally, for every data, p different time series (basis output) are stored in a separate $\boldsymbol{\beta}$ matrix of size $N \times p$.

The reconstruction of SNR time series for every sample and each template requires large computational costs. Since we are only interested in triggers exceeding a certain threshold, it is better not to reconstruct the complete SNR time series, rather only in the vicinity of the triggers. We propose a two-step hierarchical scheme for reconstruction, which performs a coarse reconstruction, and then a finer reconstruction around the triggers obtained in the first stage. In the first stage, we consider fixed nonoverlapping bins of w samples. Then the outputs from each basis

vector β_k are averaged in the bins referred as the averaged time series

$$\beta_k^{\mathrm{avg}}[t_i] = \sum_{j=0}^{w-1} \beta_k[t_{i\times w+j}]/w, \tag{9}$$

where $i = \{0, \ldots, N/w - 1\}$ corresponds to different bins, and $j = \{0, \ldots, w - 1\}$ goes over the samples inside each bin. The bin index i can be considered as a sample point in the shortened averaged time series. By linearly combining the β_k^{avg} with the decomposition coefficients $c_{k,\zeta}$ similar to Eq. (7), results in an averaged SNR times series for the template h_ζ. We identify the first-stage triggers and their corresponding bins having the average SNR above a first-stage threshold ρ_{I}. In the next step we perform a finer reconstruction for every sample inside all the triggering bins. Triggers from the second stage which are above a second stage threshold ρ_{II} are referred as the final triggers. In the Fig. 2, we show the hierarchical reconstruction of SNR time-series around a trigger.

Our aim is to minimize the total costs for the hierarchical method without losing any sensitivity. The sensitivity is determined by computing the fraction of triggers recovered at a given SNR, or conversely, the SNR threshold at which all the triggers are recovered. We compute the sensitivity (using the latter approach) by comparing the hierarchical distribution of the triggers with the original distribution. The parameters w, ρ_{I} are fine tuned under the constraint of reaching a fixed target SNR (ρ_{target}) to optimize the total costs. We give a detailed description to compute ρ_{target} in Sec. III F.

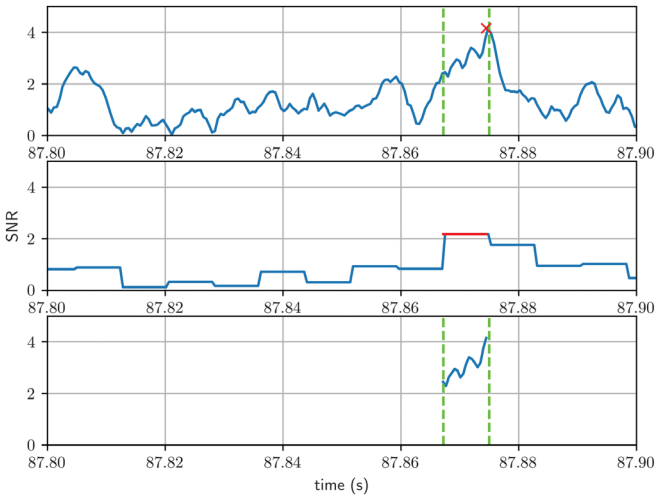

FIG. 2. Demonstration of the hierarchical scheme, the first panel shows the reconstructed SNR time series without any averaging and a single trigger (SNR ≥ 4). In the middle panel is the averaged SNR series obtained from binned averaging of $w = 16$ samples (shown in green in the first panel), and the red region corresponds to the bin containing the original trigger. The last panel shows a finer reconstruction performed only for the bin having average SNR greater than the first stage threshold.

D. Fast first stage filtering using templates

To reduce the first-stage costs the average SNR can be obtained by the faster template method which is mathematically equivalent to reduced-basis filtering. Using the template method, the average SNR time series can be obtained by binned averaging of the integrand in the Eq. (4) and then performing an IFFT of shorter length. To demonstrate the averaging process mathematically, we consider a single bin b with w samples, and compute the averaged SNR for the same, which is denoted by $\langle \rho_\theta(t) \rangle_b$. The w time samples in the bin are represented by $t = wb + r$, where $r \in [0, w - 1]$. We then write the discretized version of the Eq. (4) averaging the SNR time series over w samples

$$\langle \rho_\zeta(t) \rangle_b = \frac{1}{Nw} \sum_{r=0}^{w-1} 4\Delta f \sum_{f=0}^{N-1} \frac{\tilde{s}[f]\tilde{h}_\zeta^*[f]}{S_n[f]} e^{2\pi i f(wb+r)/N}. \tag{10}$$

Now, breaking the summation over $f \in [0, N - 1]$ into a double sum by expressing $f = lN/w + f'$, we can rewrite the single summation of any function $\tilde{g}(f)$ in the Fourier domain as

$$\sum_{f=0}^{N-1} \tilde{g}[f] = \sum_{f'=0}^{N/w-1} \sum_{l=0}^{w-1} \tilde{g}\left[l\frac{N}{w} + f'\right], \tag{11}$$

where $l \in [0, w - 1]$ and $f' \in [0, N/w - 1]$. This modifies Eq. (10) to

$$\begin{aligned} \langle \rho_\zeta(t) \rangle_b &= \frac{4w\Delta f}{N} \sum_{f'=0}^{N/w-1} e^{2\pi i f' \frac{w}{N} b} \\ &\quad \times \underbrace{\frac{1}{w^2} \sum_{l=0}^{w-1} \frac{\tilde{s}[l\frac{N}{w} + f']\tilde{h}_\zeta^*[l\frac{N}{w} + f']}{S_n[l\frac{N}{w} + f']} \sum_{r=0}^{w-1} e^{2\pi i (l\frac{N}{w} + f')\frac{r}{N}}}_{=\Omega(f')} \\ &= \frac{4w\Delta f}{N} \sum_{f'=0}^{N/w-1} e^{2\pi i f' \frac{w}{N} b} \Omega(f'). \end{aligned} \tag{12}$$

To simplify Eq. (12), we introduce a new frequency series $\Omega(f')$ which represents the binned average of the PSD-weighted correlation of the data and template. The second summation term in $\Omega(f')$ has a closed-form solution and can be computed analytically. We notice that the last line of Eq. (12) has a similar form to Eq. (4) but, the integrand replaced with $\Omega(f')$ having only N/w number of samples. Suggesting that the average SNR time series can be obtained by an IFFT of $\Omega(f')$, where one can think of b as the new time equivalent variable which ranges between $[0, N/w - 1]$. Therefore, it is possible to perform the first

stage using basis or templates, however, the second-stage filtering needs to be done with the basis.

E. Implementation

We now discuss the implementation of our method which is divided into two parts: 1) preparation and 2) matched filtering. The preparation stage is implemented partially on GPUs, whereas matched filtering is performed entirely on the GPUs. We use several Nvidia GV100 GPUs, each having memory of 32 GB as well as several RTX 2070 Super each with 8 GB of memory. Our code is written in C language and uses various optimized libraries for different purposes. Operations on the GPU are performed using CUDA [71] an application programming interface by Nvidia.

In the preparation stage, we perform PCA on the template bank to obtain the reduced basis and the respective decomposition coefficients. Matrix multiplications on the GPUs in this stage are performed using the *cuBLAS* library from CUDA. We begin by computing the covariance matrix in several parts in parallel using cuBLAS. Afterwards, we combine all the parts to obtain the final matrix $\mathbf{C}$. The Lanczos algorithm for EVD of $\mathbf{C}$ is implemented on CPUs using the SLEPc [72] and PETSc [73] libraries. We obtain the decomposition coefficients matrix $\mathbf{D}$ by multiplying the matrices $\mathbf{T}$ and $\mathbf{P}$. The preparation stage is computed in advance and performed only once. Results from this stage—the basis and the decomposition coefficients—are stored on hard drives for the later matched-filtering stage. To reduce the input/output (IO) bandwidth, we compress the PCA results before writing them on the hard drive.

In the next stage, we read the output from the previous stage to match filter data using our hierarchical scheme. To reduce the time-intensive memory transfers between CPU and GPU, we load the matrices $\mathbf{P}$ and $\mathbf{D}$ at the same time on to the GPUs. We divide the data into several smaller segments such that we can optimally utilize the memory of the GPUs while filtering each segment in parallel. We use data segments of 128s and a sampling rate of 2048 Hz. Each data segment has $N = 128 \times 2048$ samples which overlap with $N/2$ samples from the previous segment. We employ the cuFFT library from CUDA to perform the FFTs in this stage. Using cuFFT we perform FFT for a batch of data segments in parallel. In the next step, we interpolate the basis vectors and multiply them with $\tilde{s}$ along with $S_n(f)^{-1/2}$. Afterward we perform in-place batched IFFTs to obtain the filtering output from the basis. The in-place technique saves GPU memory by recycling the allocated input memory to write the output.

To perform the hierarchical reconstruction we first average the output from basis to obtain the $\boldsymbol{\beta}^{\mathrm{avg}}$ matrix. The first stage reconstruction is done using cuBLAS by multiplying $\boldsymbol{\beta}^{\mathrm{avg}}$ and $\mathbf{D}$, which outputs the average SNR time series. Next, we use a dedicated function on the GPU to find triggers with average SNR above ρ_{I}. Once the triggers are identified, we store their bin indices along with their corresponding average SNRs. Since these triggers are not contiguous in memory, further reconstruction of the triggering bins cannot be performed by simple matrix multiplication. Therefore, instead of using the optimized cuBLAS library, we use a custom-built function on the GPU to perform the second reconstruction.

F. Cost estimation

In this subsection we estimate the floating-point operations required by the two different matched-filtering schemes. The purpose of estimating the number of operations is to get a rough idea of the scaling relations involved for the total filtering costs. Moreover, it will allow us to estimate the improvement in performance due to the proposed hierarchical scheme. In both methods, we split the data into blocks of N samples, having an overlap of $N/2$ samples with the previous block. This is generally done to avoid corrupt SNR samples at the start and end of a data segment [29]. Hence, filtering a single block results in $N/2$ unique SNR time samples. Most operations involve complex numbers unless otherwise specified. Throughout the cost estimation, we consider six operations for multiplication and two operations for the addition of two complex numbers. To estimate the costs for the FFTs we consider a split-radix method [74].

We first estimate the costs for the template method which will be our baseline comparison. The first step is to compute the forward real-to-half-complex FFT of the data with N samples, which requires $3/2N \log N$ operations per block [74]. Computing the integrand of the matched filtering Eq. (1) for T templates requires $6NT$ operations. Finally, an inverse complex-to-complex FFT is required to obtain the SNR time series for each template, and this attributes to $5TN \log N$ operations, where each IFFT requires $5N \log N$ operations. In total the template method for a single block requires $N \log N(3/2 + 5T) + 6TN$ operations. Usually, the number of templates is huge ($T \gg 1$), so we can neglect the cost for the forward FFT of data. Therefore, the total floating-point operations z_{basic} for filtering $N/2$ data samples with the template method can be approximated to

$$z_{\mathrm{basic}} = NT(5 \log N + 6). \tag{13}$$

Now, we estimate the costs for the two-stage hierarchical filtering. As shown in the Secs. III C and III D, the first stage can be performed either by using the basis or the templates. We evaluate the first-stage costs by considering the faster template method described in the Sec. III D starting with the forward FFT of the data which needs $3/2N \log N$ operations. The weighted correlation of the matched filter in Eq. (12) is then obtained for every template in $6NT$ multiplicative operations. Then, we perform binned averaging of the correlations to get reduced

frequency series $\Omega(f')$ of size N/w for each template, and this requires $2NT/w$ operations. For every template we obtain the average time series by computing the IFFT of $\Omega(f')$ in $5NT/w\log(N/w)$ operations. Hence, the number of floating-point operations (neglecting the forward FFT) required for the first stage is

$$z_{\text{first}} = NT\left(\frac{5}{w}\log\left(\frac{N}{w}\right) + 6 + \frac{2}{w}\right). \tag{14}$$

In the next stage, we compute a finer reconstruction of w points around each first stage trigger. The costs for the second stage are calculated in terms of the number of first stage triggers. We denote the number of first-stage triggers for a single template by $f(w,\rho_{\text{I}})$. We assume that the number of triggers do not vary for different templates, and thus, can be obtained from a single test template. This assumption is justified for the templates in the vicinity of the test template as the number of triggers would be roughly the same. In addition, since $f(w,\rho_{\text{I}})$ decreases rapidly at higher SNRs, the error in the total costs due to our assumption is negligible. Since the fast first stage does not involve computing the basis outputs β_k, we evaluate them in the second stage in $Np(5\log(N)+6)$ operations. Using the above assumption, the second stage requires $z_{\text{second}} = 4pwf(w,\rho_{\text{I}})T + Np(5\log(N)+6)$ operations. Summing up the costs from both the stages, the total floating-point operations required for the hierarchical method are

$$z_{\text{total}} = NT\left(\frac{5}{w}\log\left(\frac{N}{w}\right) + 6 + \frac{2}{w}\right) + (4pwf(w,\rho_{\text{I}})T + Np(5\log(N)+6)). \tag{15}$$

The final costs in Eq. (15) are obtained in terms of two nuisance parameters w and ρ_{I}. An appropriate choice of the first-stage threshold ρ_{I} is important in determining the background trigger distribution. (This is to not miss any potential triggers in the first stage, because only triggers that are followed till the second stage are accounted in the background estimation.) To ensure that we recover all the triggers using a reliable first-stage threshold in the two-stage filtering scheme, we compare the hierarchical distribution of second-stage triggers against the trigger distribution from the flat scheme. The idea is to identify a target SNR (ρ_{target}) as a function of (w,ρ_{I}), such that the hierarchical scheme recovers 99% of the total triggers from the flat scheme. Considering a certain first stage configuration given by specific values of (w,ρ_{I}), we denote the number of final triggers above ρ_{II} as $n_{\text{final}}(\rho_{\text{II}})$. Similarly, using the same threshold ρ_{II}, we denote triggers from the flat scheme as $n_{\text{flat}}(\rho_{\text{II}})$. The target SNR ρ_{target} is then defined as

$$\rho_{\text{target}}(w,\rho_{\text{I}}) = (\min(\rho_{\text{II}})|n_{\text{final}}(\rho_{\text{II}}) \geq 0.99 n_{\text{flat}}(\rho_{\text{II}})). \tag{16}$$

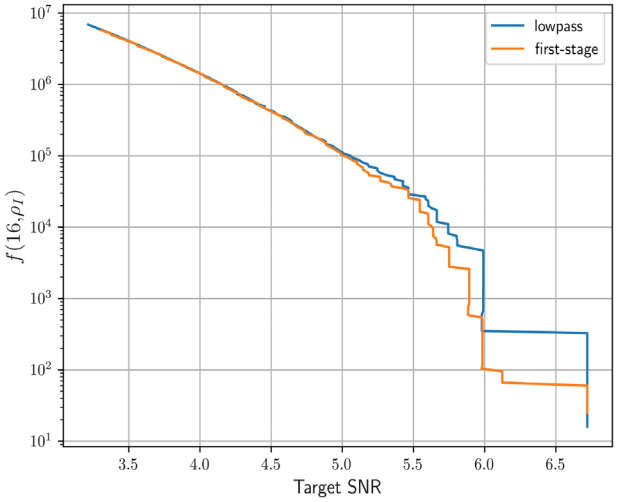

FIG. 3. Comparing two different coarse filtering methods—low-pass (blue) and first-stage (orange). We show the number of first-stage triggers $f(16,\rho_I)$ for simulated Gaussian noise as a function of target SNRs. The two methods have broadly the same performance except for SNRs > 5 where the low-pass method registers up to 10 times more false alarms.

G. Low-pass filter interpretation for the first stage

There are many ways to coarse filter the data to produce a reduced SNR time series; one such methods is using a low-pass filter combined with decimation. Our proposed method of averaging the SNR time series (described in Sec. III D) resembles, qualitatively, a low-pass filter, however it takes into account the entire frequency range as seen in Eq. (11).

We test the performance of a low-pass and our first-stage filtering methods by filtering simulated Gaussian noise. A metric for comparison could be the number of false alarms produced for a given target SNR. Since the target SNR depends on the thresholding criterion we obtain the thresholds respectively for each method. Using Eq. (16), we obtain the number of first stage triggers $f(16,\rho_{\text{I}})$ as a function of the target SNR as shown in the Fig. 3. We observe that both methods have very similar, but not identical, performance. Both methods produce the same number of triggers at SNRs ≤ 5, but for SNRs > 5 the low-pass method triggers more false alarms than the first stage—we observe up to ten times more false alarms. This indicates that because the first-stage filter preserves the high-frequency content, it has slightly better sensitivity over a low-pass filter.

IV. ACCURACY AND PERFORMANCE ANALYSIS OF THE HIERARCHICAL METHOD

In the previous section we demonstrated our method and estimated the required costs for filtering. In this section we present the accuracy of the hierarchical filtering results and assess the reduction in the required number of operations

with respect to the baseline i.e., the template method. Furthermore, we measure the gain in performance by implementing matched filtering on GPUs relative to the established CPU implementations.

We use the subregion (shown in Fig. 1 in orange) to demonstrate our method. This subregion covers the parameter ranges $M \in [5.72, 12.05]$ and $q \in [1.0, 11.05]$. Using a tolerance of 10^{-5} as per Eq. (8), we obtain $p = 254$ for this subregion. We want to estimate a conservative reduction in the total costs that scale linearly with p. Following this reason, we choose the mentioned subregion as it corresponds closely to the average $\langle p \rangle$.

Current offline and online searches [51,75] generally use a SNR threshold of ∼4–5 for the single-detector triggers. However, searches involving a large number of templates are affected by increased background due to noise triggers [40,48], and thus, higher SNR thresholds are used to detect events at a constant false alarm rate. These kinds of search scenarios also happen to be the case where cost-efficient algorithms are necessary. Therefore, in this work, we target SNR thresholds of 5 and above.

A. Accuracy of the SNR

We expect two primary contributions to the SNR loss in our method—*truncation of the number of eigenvalues* and *interpolation of the basis*. Error due to truncating the eigenvalues is translated as the SNR loss via the Eq. (8), and is regulated by choosing an appropriate number of basis vectors. The loss due to linear interpolation of the basis is quantified in terms of the mismatch between the interpolated and fully sampled templates. We evaluate the total loss by computing the relative error in the SNR time series obtained using our method and the template method. We filter simulated colored Gaussian noise from the PSD to acquire the SNR time series. Comparing triggers from every template, we note the maximum relative error in the SNR values and plot it against the SNR thresholds as shown in Fig. 4.

Based on our sensitivity requirements we are interested only in SNR = 5 and above, and from Fig. 4 we observe the relative error to be ∼0.4% for SNR = 5.0. This amount of error can be tolerated because the observed loss is less than the error due to mismatch of the templates, which is up to 3%. We also notice that the relative error decreases even further with higher SNR thresholds. Therefore, it is justified that our method successfully recovers the SNR values for search scenarios requiring SNR thresholds ≥ 5.

B. Comparing performance with template-based matched filtering

Our hierarchical method is characterized by two parameters, the averaging bin size w, and the first-stage threshold ρ_{I}. For testing the hierarchical scheme, we use four different averaging bin sizes $w \times dt$ with $w \in [2, 4, 8, 16]$, and various different values of ρ_{I}. We drop the dt factor when referring w, to suggest to the reader that we simply average over w samples. We test our scheme using noise generated from the aLIGO PSD sampled at 2048 Hz. We also check our method for a population of BBH signals within the case study subregion. Finally, we compare the estimated costs required by the hierarchical method against the template method. We use PyCBC software library [76] to generate simulated data containing Gaussian noise and to perform injections.

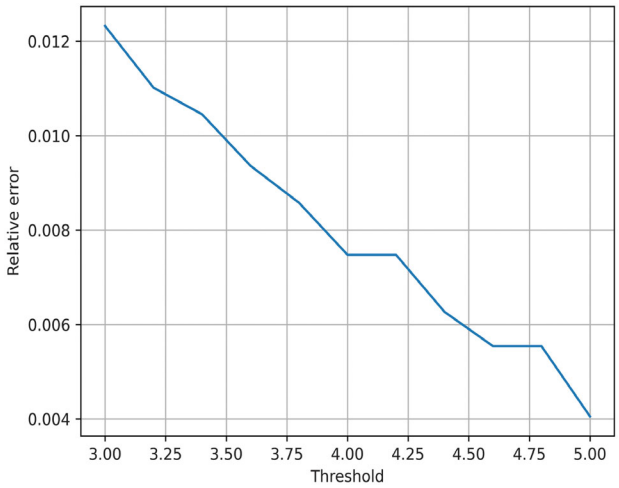

FIG. 4. Maximum relative error between the original and reconstructed SNR values as a function of SNR threshold. The incurred error scales inversely with the SNR thresholds and is smaller than the error due to the discreteness of the templates for relevant SNR thresholds.

1. Number of required operations

Using the estimates in Sec. III F, we compare the number of operations required by the hierarchical scheme with the template method. We begin by testing only Gaussian noise generated using the aLIGO PSD. For this purpose, we generate a total of ∼7.4 days of data using different seeds sampled at 2048 Hz. For filtering purposes, the simulated data is then divided in smaller segments of 128s with an overlap of 64s from the previous segment.

To estimate the total costs for the hierarchical method Eq. (15), we first determine the number of first-stage triggers $f(w, \rho_{\mathrm{I}})$ by varying the hierarchical parameters. We neglect the variation of $f(w, \rho_{\mathrm{I}})$ for different templates for reasons discussed previously in Sec. III F. $f(w, \rho_{\mathrm{I}})$ is obtained simply by iterating over different values of the first stage threshold ρ_{I} for different (fixed) w and by counting the total first stage triggers above the same threshold. The observed $f(w, \rho_{\mathrm{I}})$ with respect to ρ_{I} for different values of w is shown in Fig. 5.

It is seen from Fig. 5 that the number of first stage triggers decreases with increasing ρ_{I}, as expected. But more interestingly, we notice that fewer triggers are recovered when the averaging is done over more samples for the same

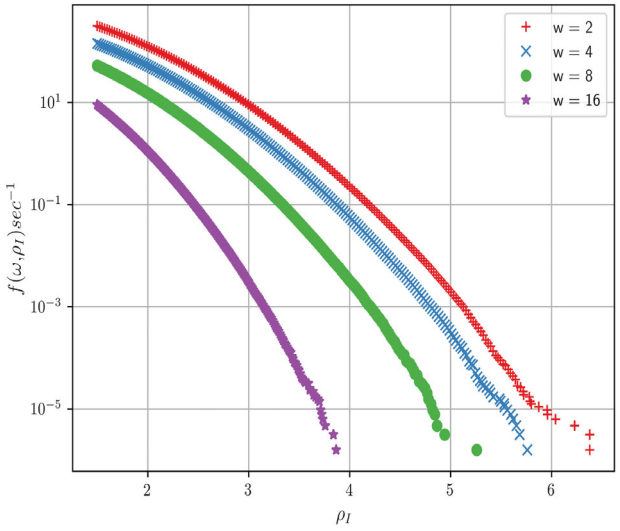

FIG. 5. The number of first stage triggers $f(\omega, \rho_I)$ per second as a function of ρ_I for different averaging bin sizes w (shown in the legend). As w is increased, ρ_I must be decreased to recover the same number of triggers f.

ρ_I, i.e., bigger w. Therefore, to maintain the same sensitivity or the number of final triggers while w increases, the first-stage threshold must be lowered.

A particular combination of the parameters (w, ρ_I) determines a specific target SNR ρ_{target} for the hierarchical method without any loss in sensitivity as discussed in Sec. III F. The target SNR is computed for a fixed w and different values of ρ_I, by iterating over different values of second stage SNR ρ_{II} using Eq. (16). In Fig. 6 we plot the relationship between ρ_I and ρ_{target} for different w. Using the

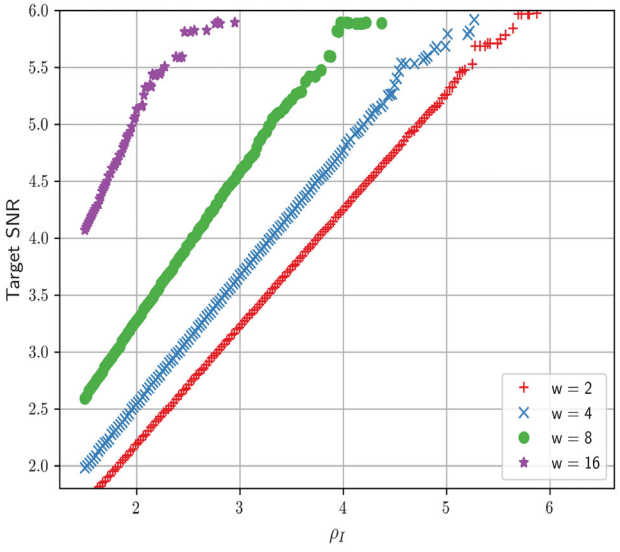

FIG. 6. The plot shows different combinations of w and ρ_I to achieve a specific target SNR ρ_{target} without losing any sensitivity. The choice of w and ρ_I determines the total cost of the hierarchical method for the corresponding ρ_{target}.

results obtained in Fig. 6 we can choose various combinations of w and ρ_I to reach a desired ρ_{target}. We notice from the plot that to reach the same ρ_{target}, bigger w requires a lower value of ρ_I. This information combined with the previous plot suggests that for a specific ρ_{target}, choosing a bigger w leads to more number of first stage triggers.

Now, we combine the results from Figs. 5 and 6 to estimate the final costs Eq. (15) in terms of ρ_{target} as shown in the Fig. 7. These costs are normalized by the costs required by the template method Eq. (13) (shown by the horizontal orange line). To demonstrate the contributions in the total costs from the individual stages separately, we plot the first stage and the total costs together. The first-stage costs are constant with respect to ρ_{target} and scale inversely with w. On the other hand, due to the rapid increase of the first-stage triggers $f(w, \rho_I)$ at low SNRs, the second-stage costs become dominant. It is also inferred from the figure that the second stage costs are more for larger w at a constant ρ_{target}. We notice from Fig. 7, the total costs converge to the first stage costs at higher SNRs, and hence, infer that the first stage leads to the dominant costs for the hierarchical matched filtering.

From Fig. 7 we observe a reduction in total costs compared to the baseline for all choices of w complying to our desired target of $\rho_{\text{target}} \geq 4.0$. We notice for $\rho_{\text{target}} = 5.0$, the setting with $w = 8$ achieves a relative speed up factor of 6, which corresponds to a reduction of ∼83% in the total costs. It is observed that the hierarchical method performs better with increasing SNR thresholds. Our method achieves the best computational gain of ∼10 times which is equivalent to a reduction of ∼90% in the total costs, for $\rho_{\text{target}} = 6.0$ using $w = 16$. We have not tested the scheme for higher values of w, but by extrapolating the obtained results we may infer that the hierarchical method might perform better with $w > 16$ for even higher SNR thresholds.

While estimating the target SNR we use a fixed value of the recovery ratio = 99% [see Eq. (16)]. To understand the impact of varying the recovery ratio, we compare the total costs for three values of the recovery ratio: 0.995, 0.99, and 0.9. We plot the total costs as a function of target SNR for all averaging window sizes in the Fig. 8. From the plot we observe that there is no significant change in the total costs at SNRs ≥ 6 and for smaller windows. But we notice a further reduction in costs for larger windows, especially at low SNR values, in exchange for the reduced accuracy. We infer from Fig. 8 a general trend for the total cost that is proportional to the recovery ratio; the curve translates to left or right, respectively. Moreover, the shift increases with the window size. This suggests that our results are not sensitive to small changes in the recovery ratio when close to unity. Depending on the search requirements, the recovery ratio is another parameter for further tuning performance.

To ensure that higher SNR triggers are recovered using only ρ_I, and also to sanity check our method for recovering

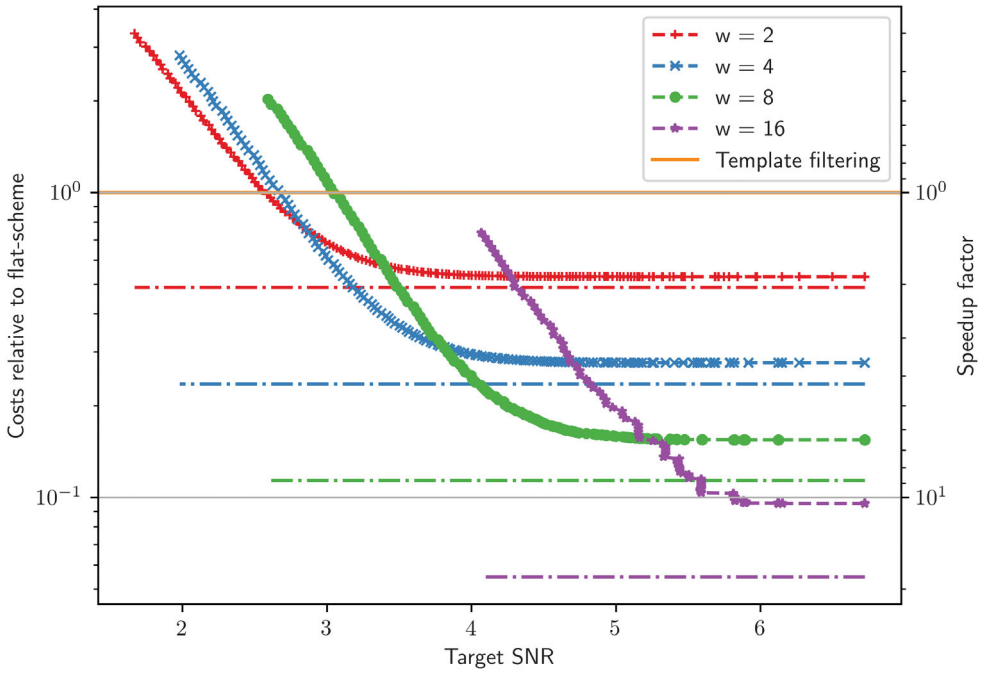

FIG. 7. The computational cost (in terms of FLOP) for hierarchical filtering ∼7.4 days of simulated data containing only Gaussian noise against 6250 templates. We show the costs for different (fixed) w which are normalized by the direct template filtering cost (orange line) versus the target SNR. The first stage costs are indicated by dot-dash lines, and the lines with markers correspond to the total costs. On the right, the vertical axis shows the relative speed-up factor (log scale) compared to the template method. The best speed up for a desired ρ_{target} is achieved by choosing w accordingly from this plot.

CBC signals. We test our scheme for a population of 1000 CBC injections randomly generated within the case study region. For every injection, we create separate strains of 128 sec and place signals randomly into simulated data containing only colored Gaussian noise. The signals correspond to SNRs ranging between [4.5, 30]. We use two test cases of $\rho_{\text{target}} = 4.0$ and 6.0 using $w = 16$. Using the appropriate first stage cutoff ρ_{I} from Fig. 6 we recover all the injections that are above the respective test cases of SNR threshold. Therefore, we verify the reliability of our method to recover CBC signals using only ρ_{I} even at higher SNRs.

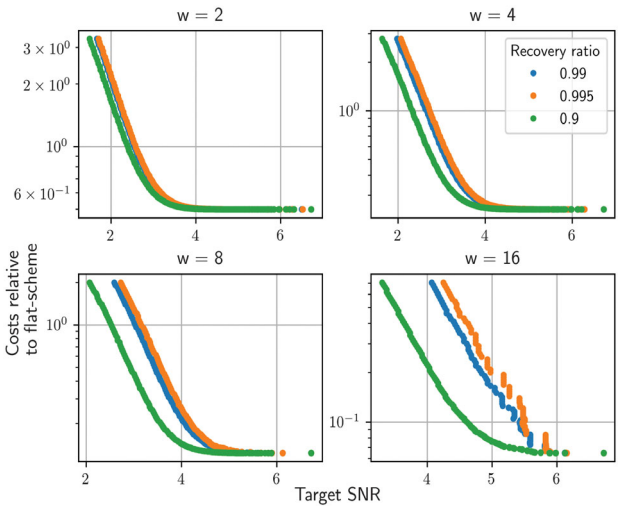

FIG. 8. Varying the recovery ratio used in the target SNR. We use three values of the recovery ratio 0.99 (blue), 0.995 (orange), and 0.9 (green) to test its impact on the total costs. We show the total relative costs versus the target SNR for four different window sizes. We notice the total costs are not sensitive for small changes in the recovery ratio, but can be further fine tuned using this ratio as a parameter.

We now make a few key remarks. The total costs of the hierarchical method are dominated by the first stage at higher SNR thresholds, which can be reduced by choosing a bigger w. On the other hand, the second stage costs are dominant at lower SNRs which grow as w increases. As seen in Fig. 7, for different SNRs we obtain different optimal window lengths, e.g., for $\rho_{\text{target}} = 5.0$ the best setting is when $w = 8$. Depending upon the threshold criterion required for the search the optimal choice of w can be chosen based on Fig. 7. In the case where the search demands higher SNR thresholds than shown in the plot, a larger w may be preferred to further reduce the costs.

2. Observed performance

In this section we measure the performance of matched-filtering implementations on GPUs to estimate a realistic improvement compared to the established search pipelines. For this purpose, we use a widely quoted performance

TABLE II. Comparing different implementations of the matched filtering schemes on GPUs (first and second row) with the established PyCBC schemes on CPUs (third and fourth row). In the second column we show the throughput for the respective methods, and in the third and fourth columns are the throughput per euro and per watt of power consumption respectively. The expected peak performance of the hierarchical method is estimated for SNR $= 5.0$ in the second row.

Method	Throughput	Throughput/Euro	Throughput/W
cuFFT (in-situ)	4000×10^3	400	14×10^3
Hierarchical scheme (expected)	2300×10^4	2300	82×10^3
PyCBC live	6300	17	31
PyCBC offline	12,000	32	60

metric—the *throughput* of a search method—used for determining the number of templates analyzed in real time. Consider a data segment of N secs filtered against T templates and the filtering process takes t seconds, then the throughput would be NT/t templates processed in real time. Once again, we filter simulated colored Gaussian noise sampled at 2048 Hz for 64 seconds to evaluate this metric.

We benchmark the template method implemented using the optimized *cuFFT* library from CUDA. The template method *in situ* took roughly 100 ms to filter 64s of data against 6250 templates per GPU. Therefore, achieving an *in situ* performance of 4000×10^3 templates processed in real time on a single Nvidia GV100. We want to remark that the second-stage reconstruction is not optimized and thus, we could not benchmark the hierarchical scheme to its full potential. Considering the costs from Fig. 7, we estimate the expected peak performance of a completely optimized hierarchical method, which suggests that the hierarchical implementation may require only 12 ms to perform the cuFFT equivalent filtering. Hence, we expect an increase of roughly an order of magnitude in the throughput (second row in Table II) if the second stage is fully optimized. Work is in progress for optimizing the second stage.

We now compare performances using previously quoted numbers from currently used search pipelines—PyCBC live [75] and PyCBC offline [51]—as shown in Table II. The PyCBC search methods are implemented on multiple CPU cores, whereas ours are on multiple GPUs. Depending upon the search method the throughput is standardized by templates processed in real time per core (GPU) for PyCBC (hierarchical) implementation. We notice from the first column in Table II that using the latest GPUs gives an enormous improvement in the throughput. However, the PyCBC numbers are not quoted from an up to date hardware implementation and hence, a fair comparison might require the latest hardware.

Furthermore, we also present commercially motivated metrics to benchmark the performance, measuring the cost and energy efficiency of hardware while filtering. These metrics are computed by normalizing the throughput by the total cost or the energy consumption of the hardware respectively. The two metrics for the different search schemes are listed in Table II.

It is apparent from Table II, that due to GPU's ability to perform tasks in a highly parallelized fashion, GPUs can analyze $\sim 10\times$ templates more than CPUs for the same costs. We also notice that considering the same power consumption a single GPU is equivalent to $\sim 10^3$ CPU cores for analyzing templates at a given instant. These metrics suggest that GPUs are energy and cost efficient in performing matched filtering, which motivates their application in CBC searches. In addition, proper implementation of our hierarchical method will allow further improvement in the efficiency of the hardware and will help reduce the time required to perform extensive offline searches.

V. DISCUSSION AND FUTURE PROSPECTS

In this work we have demonstrated, using simulated data containing Gaussian noise, an efficient way of matched filtering. We filter using a reduced basis and employ a new hierarchical method to reduce the reconstructions costs. Compared to the template based filtering, our method is $\sim 10\times$ faster than the template-based filtering methods without losing sensitivity at a threshold SNR $= 6$, and $\sim 6\times$ for SNR $= 5$. The gain in performance increases with higher SNR thresholds and is currently estimated for a specific region of the parameter space. Our method is successful in recovering the original flat search background, and thus, does not compromise the significance of detected candidates with SNR above the SNR threshold.

We demonstrate the advantages of implementing matched filtering methods on the latest GPUs. We compare the throughput of GPU implementation of matched filtering with the CPU implementation of current methods. Benchmarking the *in situ* performance of template-method implementation on GPUs, we observe a performance gain of 2–3 orders in magnitude compared to the PyCBC search pipelines. Our results indicate a significant improvement in performance, which may motivate the development of a fully optimized second-stage reconstruction. In addition, we present two new metrics to compare the performance of the matched-filtering implementation on different hardware. Analyzing these metrics suggests that GPUs are more cost and energy efficient in performing matched filtering than CPUs. Hence, the utilization of GPUs is encouraged for current or future searches.

A possible avenue to improve the described method would be to find better ways of performing the first stage and a faster implementation of the second stage. In this work, we use a constant sampling rate for matched filtering. Multirate sampling can be implemented to further improve the performance of the hierarchical method for cases where latency is a strong requirement or the duration of signals is significantly longer than those tested here.

In the near future, detectors will become more sensitive and thus the cost-effective hierarchical method proposed here can be useful for exploring subsolar regimes or searching for low-frequency long duration signals. Our method might play a role in reducing the computational costs for the future 3G detectors where the template bank size can be at least an order of magnitude larger [48,77] than the current CBC banks. Furthermore, this method can also be employed in new regions of the parameter space to perform computationally intensive searches for sources exhibiting precession or eccentricity.

ACKNOWLEDGMENTS

We thank Badri Krishnan, Kipp Cannon, and Tom Dent for the valuable discussions. We also thank Marlin Schäfer and Yifan Wang for their comments on the manuscript. We acknowledge the Max Planck Gesellschaft and the Atlas cluster computing team at Albert-Einstein Institute (AEI) Hannover for support.

[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), GW150914: The Advanced LIGO Detectors in the Era of First Discoveries, Phys. Rev. Lett. **116**, 131103 (2016).
[3] F. Acernese *et al.* (VIRGO Collaboration), Advanced Virgo: A second-generation interferometric gravitational wave detector, Classical Quantum Gravity **32**, 024001 (2015).
[4] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs, Phys. Rev. X **9**, 031040 (2019).
[5] Alexander H. Nitz, Thomas Dent, Gareth S. Davies, Sumit Kumar, Collin D. Capano, Ian Harry, Simone Mozzon, Laura Nuttall, Andrew Lundgren, and Márton Tápai, 2-OGC: Open Gravitational-wave Catalog of binary mergers from analysis of public Advanced LIGO and Virgo data, Astrophys. J. **891**, 123 (2020).
[6] R. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run, Phys. Rev. X **11**, 021053 (2021).
[7] Alexander H. Nitz, Collin D. Capano, Sumit Kumar, Yi-Fan Wang, Shilpa Kastha, Marlin Schäfer, Rahul Dhurkunde, and Miriam Cabero, 3-OGC: Catalog of gravitational waves from compact-binary mergers, Astrophys. J. **922**, 76 (2021).
[8] 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).
[9] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW190425: Observation of a compact binary coalescence with total mass $\sim 3.4\ M_{\odot}$, Astrophys. J. Lett. **892**, L3 (2020).
[10] R. Abbott *et al.* (LIGO Scientific, KAGRA, and VIRGO Collaborations), Observation of gravitational waves from two neutron star–black Hole coalescences, Astrophys. J. Lett. **915**, L5 (2021).
[11] B. S. Sathyaprakash and B. F. Schutz, Physics, Astrophysics and cosmology with gravitational waves, Living Rev. Relativity **12**, 2 (2009).
[12] Charalampos Markakis, Jocelyn S. Read, Masaru Shibata, Koji Uryu, Jolien D. E. Creighton, John L. Friedman, and Benjamin D. Lackey, Neutron star equation of state via gravitational wave observations, J. Phys. Conf. Ser. **189**, 012024 (2009),
[13] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), The rate of binary black hole mergers inferred from advanced LIGO observations surrounding GW150914, Astrophys. J. Lett. **833**, L1 (2016).
[14] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Binary black hole population properties inferred from the first and second observing runs of advanced LIGO and advanced Virgo, Astrophys. J. Lett. **882**, L24 (2019).
[15] Michele Maggiore *et al.*, Science case for the Einstein Telescope, J. Cosmol. Astropart. Phys. 03 (2020) 050.
[16] D. Reitze *et al.*, Cosmic explorer: The U.S. contribution to gravitational-wave astronomy beyond LIGO, Bull. Am. Astron. Soc. **51**, 035 (2019), arXiv:1907.04833.
[17] The eLISA Consortium, The gravitational universe, arXiv:1305.5720.
[18] Ken K. Y. Ng, Shiqi Chen, Boris Goncharov, Ulyana Dupletsa, Ssohrab Borhanian, Marica Branchesi, Jan Harms, Michele Maggiore, B. S. Sathyaprakash, and Salvatore Vitale, On the single-event-based identification of primordial black hole mergers at cosmological distances, arXiv:2108.07276.
[19] Stanislav Babak, Jonathan Gair, Alberto Sesana, Enrico Barausse, Carlos F. Sopuerta, Christopher P. L. Berry, Emanuele Berti, Pau Amaro-Seoane, Antoine Petiteau, and Antoine Klein, Science with the space-based

interferometer LISA. V: Extreme mass-ratio inspirals, Phys. Rev. D **95**, 103012 (2017).
[20] Matthew Evans *et al.*, A horizon study for cosmic explorer: Science, observatories, and community, arXiv:2109.09882.
[21] Surabhi Sachdev *et al.*, The GstLAL search analysis methods for compact binary mergers in advanced LIGO's second and advanced Virgo's first observing runs, arXiv:1901.08580.
[22] F. Aubin *et al.*, The MBTA pipeline for detecting compact binary coalescences in the third LIGO–Virgo observing run, Classical Quantum Gravity **38**, 095004 (2021).
[23] Xiaoyang Guo, Qi Chu, Zhihui Du, and Linqing Went, GPU-optimised low-latency online search for gravitational waves from binary coalescences, in *2018 26th European Signal Processing Conference (EUSIPCO)* (IEEE, 2018), pp. 2638–2642, 10.23919/EUSIPCO.2018.8553574.
[24] J. Creighton and W. G. Anderson, Gravitational-wave data analysis, in *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis* (John Wiley & Sons, Ltd, Weinheim, 2011).
[25] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914, Classical Quantum Gravity **33**, 134001 (2016).
[26] Miriam Cabero *et al.*, Blip glitches in Advanced LIGO data, Classical Quantum Gravity **36**, 155010 (2019).
[27] Bruce Allen, χ^2 time-frequency discriminator for gravitational wave detection, Phys. Rev. D **71**, 062001 (2005).
[28] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Effects of data quality vetoes on a search for compact binary coalescences in advanced LIGO's first observing run, Classical Quantum Gravity **35**, 06501 (2018).
[29] Bruce Allen, Warren G. Anderson, Patrick R. Brady, Duncan A. Brown, and Jolien D. E. Creighton, FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries, Phys. Rev. D **85**, 122006 (2012).
[30] Sebastian Khan, Katerina Chatziioannou, Mark Hannam, and Frank Ohme, Phenomenological model for the gravitational-wave signal from precessing binary black holes with two-spin effects, Phys. Rev. D **100**, 024059 (2019).
[31] Alejandro Bohé, Lijing Shao, Andrea Taracchini, Alessandra Buonanno, Stanislav Babak *et al.*, Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors, Phys. Rev. D **95**, 044028 (2017).
[32] Jonathan Blackman, Scott E. Field, Mark A. Scheel, Chad R. Galley, Christian D. Ott, Michael Boyle, Lawrence E. Kidder, Harald P. Pfeiffer, and Béla Szilágyi, Numerical relativity waveform surrogate model for generically precessing binary black hole mergers, Phys. Rev. D **96**, 024058 (2017).
[33] Soumen Roy, Anand S. Sengupta, and Nilay Thakor, Hybrid geometric-random template-placement algorithm for gravitational wave searches from compact binary coalescences, Phys. Rev. D **95**, 104045 (2017).
[34] Soumen Roy, Anand S. Sengupta, and Parameswaran Ajith, Effectual template banks for upcoming compact binary searches in Advanced-LIGO and Virgo data, Phys. Rev. D **99**, 024048 (2019).
[35] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Effects of data quality vetoes on a search for compact binary coalescences in advanced LIGO's first observing run, Classical Quantum Gravity **35**, 065010 (2018).
[36] Derek Davis *et al.* (LIGO Collaboration), LIGO detector characterization in the second and third observing runs, Classical Quantum Gravity **38**, 135014 (2021).
[37] Alexander H. Nitz, Collin Capano, Alex B. Nielsen, Steven Reyes, Rebecca White, Duncan A. Brown, and Badri Krishnan, 1-OGC: The first open gravitational-wave catalog of binary mergers from analysis of public advanced LIGO data, Astrophys. J. **872**, 195 (2019).
[38] Leone Bosi and Edward K. Porter, Data analysis challenges for the Einstein Telescope, Gen. Relativ. Gravit. **43**, 519 (2011).
[39] S. Hild *et al.*, Sensitivity studies for third-generation gravitational wave observatories, Classical Quantum Gravity **28**, 094013 (2011).
[40] Alexander H. Nitz and Yi-Fan Wang, Search for gravitational waves from the coalescence of sub-solar mass and eccentric compact binaries, arXiv:2102.00868.
[41] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Search for Subsolar Mass Ultracompact Binaries in Advanced LIGO's Second Observing Run, Phys. Rev. Lett. **123**, 161102 (2019).
[42] Yi-Fan Wang and Alexander H. Nitz, Prospects for detecting gravitational waves from eccentric subsolar mass compact binaries, Astrophys. J. **912**, 53 (2021).
[43] Mark Hannam, Modelling gravitational waves from precessing black-hole binaries: Progress, challenges and prospects, Gen. Relativ. Gravit. **46**, 1767 (2014).
[44] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Search for eccentric binary black hole mergers with advanced LIGO and advanced Virgo during their first and second observing runs, Astrophys. J. **883**, 149 (2019).
[45] Alexander H. Nitz, Amber Lenon, and Duncan A. Brown, Search for eccentric binary neutron star mergers in the first and second observing runs of Advanced LIGO, Astrophys. J. **890**, 1 (2020).
[46] Thibault Damour, Achamveedu Gopakumar, and Bala R. Iyer, Phasing of gravitational waves from inspiralling eccentric binaries, Phys. Rev. D **70**, 064028 (2004).
[47] Theocharis A. Apostolatos, Curt Cutler, Gerald J. Sussman, and Kip S. Thorne, Spin induced orbital precession and its modulation of the gravitational wave forms from merging binaries, Phys. Rev. D **49**, 6274 (1994).
[48] Ian Harry, Stephen Privitera, Alejandro Bohé, and Alessandra Buonanno, Searching for gravitational waves from compact binaries with precessing spins, Phys. Rev. D **94**, 024012 (2016).
[49] Kipp Cannon, Adrian Chapman, Chad Hanna, Drew Keppel, Antony C. Searle, and Alan J. Weinstein, Singular value decomposition applied to compact binary coalescence gravitational-wave signals, Phys. Rev. D **82**, 044025 (2010).
[50] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Low-latency gravitational-wave alerts for multimessenger astronomy during the second advanced LIGO and Virgo observing run, Astrophys. J. **875**, 161 (2019).

[51] Samantha A. Usman *et al.*, The PyCBC search for gravitational waves from compact binary coalescence, Classical Quantum Gravity **33**, 215004 (2016).
[52] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era, Classical Quantum Gravity **33**, 175012 (2016).
[53] S. Babak, R. Biswas, P. R. Brady, D. A. Brown, K. Cannon, C.D Capano *et al.*, Searching for gravitational waves from binary coalescence, Phys. Rev. D **87**, 024033 (2013).
[54] B. S. Sathyaprakash and S. V. Dhurandhar, Choice of filters for the detection of gravitational waves from coalescing binaries, Phys. Rev. D **44**, 3819 (1991).
[55] S. Babak, R. Balasubramanian, D. Churches, T. Cokelaer, and B. S. Sathyaprakash, A Template bank to search for gravitational waves from inspiralling compact binaries. I. Physical models, Classical Quantum Gravity **23**, 5477 (2006).
[56] P. Ajith, N. Fotopoulos, S. Privitera, A. Neunzert, N. Mazumder, and A. J. Weinstein, Effectual template bank for the detection of gravitational waves from inspiralling compact binaries with generic spins, Phys. Rev. D **89**, 084041 (2014).
[57] Ian W. Harry, Bruce Allen, and B. S. Sathyaprakash, A stochastic template placement algorithm for gravitational wave data analysis, Phys. Rev. D **80**, 104014 (2009).
[58] Prasanna Joshi, Rahul Dhurkunde, Sanjeev Dhurandhar, and Sukanta Bose, Optimal χ^2 discriminator against modeled noise transients in interferometric data in searches for binary black-hole mergers, Phys. Rev. D **103**, 044035 (2021).
[59] Samantha A. Usman *et al.*, The PyCBC search for gravitational waves from compact binary coalescence, Classical Quantum Gravity **33**, 215004 (2016).
[60] Shaun Hooper, Shin Kee Chung, Jing Luan, David Blair, Yanbei Chen, and Linqing Wen, Summed parallel infinite impulse response (SPIIR) filters for low-latency gravitational wave deetection, Phys. Rev. D **86**, 024012 (2012).
[61] Kipp Cannon, Romain Cariou, Adrian Chapman, Mireia Crispin-Ortuzar, Nickolas Fotopoulos, Melissa Frei, Chad Hanna, Erin Kara, Drew Keppel, Laura Liao, Stephen Privitera, Antony Searle, Leo Singer, and Alan Weinstein, Toward early-warning detection of gravitational waves from compact binary coalescence, Astrophys. J. **748**, 136 (2012).
[62] Bhooshan Gadre, Sanjit Mitra, and Sanjeev Dhurandhar, Hierarchical search strategy for the efficient detection of gravitational waves from nonprecessing coalescing compact binaries with aligned-spins, Phys. Rev. D **99**, 124035 (2019).
[63] S. D. Mohanty and S. V. Dhurandhar, A hierarchical search strategy for the detection of gravitational waves from coalescing binaries, Phys. Rev. D **54**, 7108 (1996).
[64] Anand S. Sengupta, Sanjeev V. Dhurandhar, Albert Lazzarini, and Tom Prince, Extended hierarchical search (EHS) algorithm for detection of gravitational waves from inspiraling compact binaries, Classical Quantum Gravity **19**, 1507 (2002).
[65] C. E. Shannon, Communication in the presence of noise, Proc. IRE **37**, 10 (1949).
[66] Sumeet Kulkarni, Khun Sang Phukon, Amit Reza, Sukanta Bose, Anirban Dasgupta, Dilip Krishnaswamy, and Anand S. Sengupta, Random projections in gravitational wave searches of compact binaries, Phys. Rev. D **99**, 101503 (2019).
[67] Amit Reza, Anirban Dasgupta, and Anand S. Sengupta, Random projections in gravitational-wave searches from compact binaries II: Efficient reconstruction of the detection statistic, arXiv:2101.03226.
[68] Kanchan Soni, Bhooshan Uday Gadre, Sanjit Mitra, and Sanjeev Dhurandhar, Hierarchical search for compact binary coalescences in the Advanced LIGO's first two observing runs, Phys. Rev. D **105**, 064005 (2022).
[69] Tito Dal Canton and Ian W. Harry, Designing a template bank to observe compact binary coalescences in advanced LIGO's second observing run, arXiv:1705.01845.
[70] Xuansheng Wang, Beidun Chen, Jianqiang Sheng, Hongying Zheng, Tangren Dan, and Xianfeng Wu, An improved Lanczos algorithm for principal component analysis, in *Proceedings of 2020 the 6th International Conference on Computing and Data Engineering*, ICCDE 2020 (Association for Computing Machinery, New York, NY, USA, 2020) pp. 70–74.
[71] NVIDIA, Péter Vingelmann, and Frank H. P. Fitzek, Cuda, release: 10.2.89, 2020.
[72] V. Hernandez, J. E. Roman, A. Tomas, and V. Vidal, A survey of software for sparse eigenvalue problems, Technical Report No. STR-6, Universitat Politècnica de València, 2009, https://slepc.upv.es.
[73] Satish Balay *et al.*, PETSc Web page, https://petsc.org/ (2021).
[74] Steven G. Johnson and Matteo Frigo, A modified split-radix FFT with fewer arithmetic operations, IEEE Trans. Signal Process. **55**, 111 (2007).
[75] Alexander H. Nitz, Tito Dal Canton, Derek Davis, and Steven Reyes, Rapid detection of gravitational waves from compact binary mergers with PyCBC Live, Phys. Rev. D **98**, 024050 (2018).
[76] Alex Nitz *et al.*, gwastro/pycbc:, 2021.
[77] Leone Bosi and Edward K. Porter, Data analysis challenges for the Einstein Telescope, Gen. Relativ. Gravit. **43**, 519 (2011).