Gravitational-wave template banks for novel compact binaries

Stefano Schmidt,[1,2,*] Bhooshan Gadre,[2] and Sarah Caudill[3,4]

[1]*Nikhef, Science Park 105, 1098 XG, Amsterdam, The Netherlands*
[2]*Institute for Gravitational and Subatomic Physics (GRASP),*
Utrecht University, Princetonplein 1, 3584 CC Utrecht, The Netherlands
[3]*Department of Physics, University of Massachusetts, Dartmouth, Massachusetts 02747, USA*
[4]*Center for Scientific Computing and Visualization Research,*
University of Massachusetts, Dartmouth, Massachusetts 02747, USA

(Received 3 February 2023; revised 17 November 2023; accepted 11 January 2024; published 20 February 2024)

We introduce a novel method to generate a bank of gravitational-waveform templates of binary black hole (BBH) mergers for matched-filter searches in LIGO, Virgo, and Kagra data. We derive a novel expression for the metric approximation to the distance between templates, which is suitable for precessing BBHs and/or systems with higher-order modes (HM) imprints and we use it to meaningfully define a template probability density across the parameter space. We employ a masked autoregressive normalizing flow model which can be conveniently trained to quickly reproduce the target probability distribution and sample templates from it. Thanks to the normalizing flow, our code takes a few *hours* to produce random template banks with millions of templates, making it particularly suitable for high-dimensional spaces, such as those associated to precession, eccentricity and/or HM. After validating the performance of our method, we generate a bank for precessing black holes and a bank for aligned-spin binaries with HMs: with only 5% of the injections with fitting factor below the target of 0.97, we show that both banks cover satisfactorily the space. Our publicly released code MBANK will enable searches of high-dimensional regions of BBH signal space, hitherto unfeasible due to the prohibitive cost of bank generation.

DOI: 10.1103/PhysRevD.109.042005

I. INTRODUCTION

As gravitational-wave (GW) astronomy enters a mature state, the accessible parameter space of binary black hole (BBH) mergers in LIGO [1] and Virgo [2] data continues to grow. Besides standard aligned-spin GW searches for stellar-mass BBH mergers [3–6], there are GW searches targeting the parameter space of subsolar mass black holes (BH) [7–9], primordial BHs [10], eccentric binaries [9,11–16], and intermediate-mass BHs (IMBH) [17–19]. Moreover, there is a growing interest in GW searches for more complex binaries, such as those with precession [20–26] or higher-order mode (HMs) content [21,27–30].

GW searches for signals from compact binary mergers traditionally utilize the method of matched-filtering with a template bank of model waveforms [31–36]. An optimal template bank is composed of the smallest number of templates that guarantees that only a small fraction of signal-to-noise ratio from GW signals is missed due to the discreteness of the template bank [37].

One widely used approach to bank generation—the *stochastic* method [38–40]—consists of randomly scattering templates in a defined parameter space with a rejection technique [23,41–43]. A proposed template is included in the bank only if its distance (or *mismatch*) with all the proposed templates in the bank is larger than the user-defined threshold. While this approach has proven to be very powerful, it does not scale well with (i) the number of templates and, most importantly, and (ii) with the number of dimensions of the parameter space.

Handling a large number of templates can have a large impact on computing time and memory, because for every new proposal, a waveform needs to be generated and stored and many expensive match calculations need to be performed. Furthermore, the sheer number of dimensions can have an even more catastrophic impact on the bank generation cost. Indeed, at every iteration the stochastic algorithm computes the distance between $N_\mathrm{p} \sim r^D$ pairs of templates within a given radius r. It is clear how the number of match computations diverges for large dimensional spaces.

As the BBH searches grow in complexity due to the inclusion of more physical effects and hence more dimensions, the stochastic approach struggles to produce template banks in a feasible amount of time. This poses the challenge of finding a viable alternative for template bank generation,

[*] s.schmidt@uu.nl

which is able to deliver large banks in a high-dimensional parameter space, such as those associated with precession, eccentricity, and HMs.

Revitalizing a pioneering line of research in bank generation [37,44–47], there has recently been increasing attention on *metric template placement* [48–51]. Such methods rely on approximating the distance (or mismatch) between two waveforms with a bilinear form, called metric. Although the metric is only approximate, it allows for a faster template placing, which may overcome some of the major limitations of the standard stochastic placement algorithm.

Historically, the metric was first employed to place templates on a lattice [37,44,52]. However, constructing lattice-based template banks has proven to be challenging due to the difficulties in obtaining coordinate transformation which avoids varying metric components. To overcome such difficulties, a different metric placement method, called *random*, was introduced [45] soon after. Random template banks are designed to cover the region of interest with randomly sampled templates, without any control of the template spacing. Moreover, they are not designed to cover the whole space but only a large fraction $\eta < 1$ of it (i.e., any point in space is covered with probability η).

The strength of the method is twofold: on the one hand, since no distance between templates is computed, the template placement is tremendously fast and memory efficient; on the other hand, by only covering a fraction of the space, the number of templates remains under control. Moreover, the cost does not increase for an increasing number of dimensions. While this may seem suboptimal with respect to a lattice, in [45,53,54] it is argued that for high-dimensional spaces, random template banks outperform even the best known lattice in terms of coverage (at a fixed number of templates), effectively beating the "curse of dimensionality."

Generating a random template banks requires the ability to effectively sample templates "uniformly" across the parameter space. Traditionally, due to the high dimensionality of the space, expensive sampling techniques, such as Markov Chain Monte Carlo, must be used. This poses a serious limitation to the range of applicability of the method. Without a fast sampling method, the speed up promised by the new method is washed away by the cost of a large number of metric evaluations.

In this work, we address the challenges described above by covering high-dimensional spaces with random template banks. As a first step, we derive a novel expression for the metric, which is suitable for generic precessing and/or HM waveforms. In doing so, we drop several symmetry assumptions that enters the standard metric computation. The metric is then expressed in terms of the gradients of the waveform. Second, to enable a fast template sampling, we employ machine learning and train a *normalizing flow* model to efficiently sample templates from the parameter space. While the first innovation delivers an accurate distribution for the templates throughout the space, the use of a normalizing flow allow us to generate random template banks in a few hours (including the training time).

The combination of a new metric expression and the normalizing flow model, applied to the random template placement algorithm, makes our method particularly well-suited for dealing with high-dimensional (> 4D) parameter spaces, such as those associated with precessing or eccentric searches. Our method is implemented in an open-source, production-ready, PYTHON package MBANK [55], available on GitHub[1] and on the PyPI repository.[2]

The rest of this paper is devoted to the presentation and description of our methods and software package. In Sec. II we present the details of our bank generation algorithm. In Sec. III we assess the accuracy of our template placing method in all its parts. Furthermore, we reproduce two banks available in the literature [21,56] created with independent codes: this will be the topic of Sec. IV. To demonstrate the capabilities of MBANK, in Sec. V, we present two large banks covering "exotic" regions of parameter space: a precessing bank and an IMBH bank with HM content. We also discuss some possible further applications of our normalizing flow model, including a study of the size of the precessing neutron star-black hole (NSBH) parameter space. Finally, in Sec. VI we discuss some possible future development of our work and gather some final remarks in Sec. VII.

Throughout the paper we will use the term "standard" to refer to the searches for circularized, aligned-spin BBHs without imprints of HMs, currently conducted by the LIGO-Virgo-KAGRA collaboration.

II. METHODS

When searching for a BBH signal in GW data, it is customary to use a frequentist detection statistic [21,24,57,58], which models the detector output to be composed of *gaussian* noise $n(t)$ and possibly a known GW signal $h(t)$. Given some observed data $s(t)$, the detection statistic Λ is a measure of the log probability ratio between the signal hypothesis $n + h$ and the noise hypothesis n:

$$\Lambda = \log \frac{p(s|n+h)}{p(s|n)}. \tag{1}$$

For interferometric GW observatories such as LIGO and Virgo, the observed signal takes the following form:

$$h(t) = F_+(\delta, \alpha, \Psi) h_+(t; \theta) + F_\times(\delta, \alpha, \Psi) h_\times(t; \theta) \tag{2}$$

[1] https://github.com/stefanoschmidt1995/mbank.

[2] The package is distributed under the name GW-MBANK.

The functions $F_+, F_\times$, also called antenna patterns, denote the interferometer response to the two polarizations of a GW. They depend on the sky location, parameterized by right ascension α and declination δ, and on the polarization angle Ψ. For a BBH system, the two polarizations $h_+, h_\times$ depend on two BH masses (m_1, m_2), two 3-dimensional spins ($\mathbf{s}_1$, $\mathbf{s}_2$), the inclination angle ι, the reference phase φ, the luminosity distance of the source D_L, the eccentricity e of the orbit and the mean periastron anomaly a [59].

Under the assumption of *Gaussian noise*, we can write down an explicit model for the likelihood and, after maximizing over an overall amplitude factor, Eq. (1) becomes [24,57,58]:

$$\Lambda = \frac{(\Re\langle s|h\rangle)^2}{\langle h|h\rangle} = (s|\hat{h})^2 \tag{3}$$

where we introduced a *complex* scalar product between two vectors a, b:

$$\langle a|b\rangle = 4\int_{f_{\min}}^{f_{\max}} df \frac{\tilde{a}^*(f)\tilde{b}(f)}{S_n(f)} \tag{4}$$

and the integral extends in a suitable frequency range $[f_{\min}, f_{\max}]$. In this context, $S_n(f)$ is the frequency domain autocorrelation function of the noise, also called power spectral density (PSD) and $\tilde{}$ denotes the Fourier transform. For ease of notation, we define $(a|b) = \Re\langle a|b\rangle$ and $\hat{a} = \frac{a}{(a|a)}$.

For any given observation time, a search aims to *maximize* the detection statistic Λ with respect to all the parameters of the signal model. This maximized quantity is also called signal-to-noise ratio (SNR). Depending on symmetry assumptions on the polarizations, one is able to maximize analytically over some (nuisance) parameters. For the other quantities, a brute force approach is required, where the maximized Λ is evaluated at each time on a large set of signal models, called a *template bank* [42,60]. Regardless of the nature of the signal, one is *always* able to maximize Λ over sky-location (angles α and δ), polarization angle Ψ and luminosity distance D_L, which enters as an overall amplitude scaling.

The computation of the SNR as a function of time for a single template is known as *matched filtering* and has been implemented successfully as the first stage of several pipelines to search for GW signals [34,61–68]. Modern pipelines can easily perform matched filtering on millions of templates and use the aggregated information to produce lists of GW candidates, ranked by their false alarm probability of occurrence in a noise only model.

For a circular nonprecessing signal with no HM, it holds $\tilde{h}_+ \propto i\tilde{h}_\times$ and the maximization of Eq. (3) over the nuisance parameters yields [58]:

$$\max \Lambda = \left\| \langle s|\hat{h}_+\rangle \right\|^2 = (s|\hat{h}_+)^2 + (s|\hat{h}_\times)^2 \tag{5}$$

In this simple case, $\max \Lambda$ only depends on the two BH masses m_1, m_2 and the two z-components of the spins s_{1z}, s_{2z} (4 quantities).

For the general case, where no particular symmetry is available, one obtains a different expression [21,69,70]:

$$\max \Lambda = \frac{(s|\hat{h}_+)^2 + (s|\hat{h}_\times)^2 - 2(\hat{h}_+|\hat{h}_\times)(s|\hat{h}_\times)(s|\hat{h}_+)}{1 - (\hat{h}_+|\hat{h}_\times)^2} \tag{6}$$

In this case, $\max \Lambda$ depends on 12 parameters: they are the two BH masses m_1, m_2, the two three-dimensional spins $\mathbf{s}_1$, $\mathbf{s}_2$, the inclination angle ι, the reference phase φ and the eccentricity parameters e, a. Unlike the "standard" case, an analytical maximization does not remove the dependence of ι and φ, entering in $h_+, h_\times$. Depending on the scope of a matched-filter search, a pipeline can use either Eq. (5) or (6) to filter the interferometer data with a template.

For the purpose of template placement, it is useful to think of the parameter space of BBH signals as a D-dimensional manifold $\mathcal{B}_D$, embedded in a large 12 dimensional manifold $\mathcal{B}$. Each point of the manifold corresponds to a GW signal. The number of dimensions D depends on the BBH variables under consideration. As the parameters that do not enter the interesting space can be freely neglected (i.e., set to 0 or to a meaningful constant value), the manifold $\mathcal{B}_D$ is effectively a lower dimensional *projection* of the large manifold $\mathcal{B}$.

To place templates on $\mathcal{B}_D$, it is standard to equip the manifold with a distance (called *mismatch*), which also naturally defines a volume element at every point in space. The volume element defines the "uniform" probability distribution according to the metric. A random template bank will be populated by templates drawn from such distribution, until a certain coverage is reached. For this reason, our primary concern is to sample from the manifold and to check for coverage. To effectively do so, we rely on the three steps below:

(1) Construction of a metric approximation of the match between templates. This makes $\mathcal{B}_D$ a Riemannian manifold with a volume element.
(2) Training of a normalizing flow model to sample from the manifold.
(3) Placing the templates by sampling from the normalizing flow model and checking for coverage, following [50].

The rest of this section details the steps above.

A. The metric

The definition of a metric on the manifold $\mathcal{B}_D$ provides a fast-to-compute approximation to the *mismatch* (distance) between templates and an estimation of the volume element at each point in the space.

Given two points of the manifold θ_1, θ_2, we define the overlap $\mathcal{O}(\theta_1, \theta_2, t)$ between normalized templates as:

$$\mathcal{O}(\theta_1,\theta_2,t) = \frac{1}{1-\hat{h}_{+\times}(\theta_2)^2}\{(\hat{h}_+(\theta_1)e^{ift}|\hat{h}_+(\theta_2))^2 + (\hat{h}_+(\theta_1)e^{ift}|\hat{h}_\times(\theta_2))^2 - 2h_{+\times}(\theta_2)(\hat{h}_+(\theta_1)e^{ift}|\hat{h}_\times(\theta_2)) \times (\hat{h}_+(\theta_1)e^{ift}|\hat{h}_+(\theta_2))\} \tag{7}$$

where $\hat{h}_+(\theta)e^{ift}$ is the plus polarization $\hat{h}_+(\theta)$ translated by a constant time shift t and $\hat{h}_{+\times}(\theta) = (\hat{h}_+(\theta)|\hat{h}_\times(\theta))$. The overlap amounts to the fraction of SNR recovered when filtering a signal $s = h_+(\theta_1)$ with a template evaluated at a point θ_2 using Eq. (6).

In Eq. (7), we choose to compare the h_+ polarization of the first template with both polarizations of the second template. We are forced to make such arbitrary choice by the fact that in general Eq. (6) does depend on $F_+, F_\times$. This creates an asymmetry between signal and template. Thus, if we do not want the overlap to depend on two arbitrary combination coefficients, an arbitrary choice for the signal s is needed. Of course, any linear combination of $h_+(\theta_1)$ and $h_\times(\theta_1)$ works but we set $s = h_+(\theta_1)$ for computational convenience. Numerical studies show that replacing $h_+(\theta_1)$ with any linear combination does not have a large impact on the metric definition below.

In the case of a "standard" search, $h_{+\times} = 0$ and $\tilde{h}_\times = i\tilde{h}_+$, hence the overlap simplifies to:

$$\mathcal{O}(\theta_1,\theta_2,t) = |\langle\hat{h}_+(\theta_1)e^{ift}|\hat{h}_+(\theta_2)\rangle|^2. \tag{8}$$

Note that, since Eq. (5) is symmetric[3] between signal and template, the expression for the overlap in the standard case is also symmetric. This means that an arbitrary choice on the signal composition is no longer needed, as was the case for Eq. (7).

While all the literature available [37,44–46,48,50,51] relies on the expression in Eq. (8) to derive the metric and addresses only the standard case, we tackle the general case.

Closely following [44], can maximize the overlap Eq. (7) with respect to the time shift t to obtain the *match* $\mathcal{M}(\theta_1, \theta_2)$ between templates evaluated at different points of the manifold:

$$\mathcal{M}(\theta_1,\theta_2) = \max_t \mathcal{O}(\theta_1,\theta_2,t). \tag{9}$$

The match has values in [0, 1] and trivially $\mathcal{M}(\theta_i, \theta_i) = 1$.

Even though in general the match is not symmetric and does not satisfy triangular inequality, we can use it to introduce a *distance* d between two points on the D-manifold $\mathcal{B}_D$:

$$d^2(\theta_1,\theta_2) \coloneqq 1 - \mathcal{M}(\theta_1,\theta_2). \tag{10}$$

The distance d can then by approximated locally by a bilinear form d_M:

$$d_M^2(\theta_1,\theta_2) \coloneqq M_{ij}(\theta)\Delta\theta_i\Delta\theta_j \simeq 1 - \mathcal{M}(\theta_1,\theta_2). \tag{11}$$

The bilinear form d_M is represented by a D-dimensional square matrix $M_{ij}(\theta)$, defined at each point of the manifold.

We identify $M_{ij}(\theta)$ to be the quadratic term of the Taylor expansion of $d_M(\theta + \Delta\theta, \theta)$ around $\Delta\theta \simeq 0$:

$$M_{ij}(\theta) = -\frac{1}{2}\left(H_{ij} - \frac{H_{ti}H_{tj}}{H_{tt}}\right) \tag{12}$$

where $H(\theta)$ is the Hessian of the overlap in Eq. (7), a $D+1$ square matrix. Note that the metric is positive definite (i.e., has positive eigenvalues).

A convenient expression for H in terms of the gradients of the waveform is presented in Appendix A, with the full expression given in Eqs. (A6)–(A8). While identifying the metric with the Hessian is well motivated and yields reliable results, other definitions for M_{ij} are possible; this is briefly discussed in Appendix B.

For most of the waveform models available, the gradients can be evaluated with finite difference methods. For a limited number of machine-learning based models [71–75], the gradients are available analytically.

Equipped with the metric from Eq. (12), the manifold $\mathcal{B}_D$ becomes a Riemannian manifold with line element:

$$ds^2 = M_{ij}(\theta)d\theta_i d\theta_j. \tag{13}$$

We can then use standard results from differential geometry to compute distances and volumes. In particular, the volume of a subset $\mathcal{T}$ of the manifold can be computed as:

$$\mathrm{Vol}(\mathcal{T}) = \int_{\mathcal{T}} d^D\theta\sqrt{\det M(\theta)} \tag{14}$$

where $\det M(\theta)$ is the determinant of the matrix $M_{ij}(\theta)$, also denoted as $|M|$. Moreover, we introduce the uniform probability measure, such that $p(V) \propto \mathrm{Vol}(V)$ for any $V \subseteq \mathcal{B}_D$. The measure has the following probability distribution function (PDF):

$$p(\theta) \propto \sqrt{\det M(\theta)}. \tag{15}$$

[3] Indeed, for a standard signal $s \propto h_+$, hence $\hat{s} = \hat{h}_+$, and Eq. (5) does not depend on the antenna patterns functions, if s is normalized.

Samples from the uniform distribution tend to have a "uniform" (i.e., constant) spacing, computed with the metric distance. Owing to this feature, the uniform distribution is a natural candidate to draw templates from.

B. Sampling from the manifold

To generate a random template bank, we need to sample points on the manifold $\mathcal{B}_D$ from Eq. (15). A simple way to do so is by means of a Markov Chain Monte Carlo (MCMC). However, this turns out to be unfeasibly expensive, since to obtain a single sample, the metric must be evaluated tens of times. For instance, to produce a bank with $\mathcal{O}(10^6)$ templates, $\mathcal{O}(10^7)$ metric evaluations are required.

To speed up the sampling, we introduce a *normalizing flow* model. As we will show below, in order to train the model $\mathcal{O}(10^5)$ metric evaluations are sufficient: this is a small fraction of the metric evaluations needed to run a MCMC. Once trained, the normalizing flow model produces high quality samples from Eq. (15) in a small amount of time, effectively providing templates to populate a random template bank.

A normalizing flow model [76–79] is a machine learning model widely used to reproduce and/or parametrize complicated probability distributions. Mathematically, a flow is an *invertible* parametric function ϕ_W which is trained to map samples θ from an arbitrary probability distribution $p(\theta)$ to samples $\mathbf{x}$ from a multivariate standard normal distribution $\mathcal{N}(\mathbf{x}|0,\mathbf{1})$. The space of the $\mathbf{x}$ is sometimes referred to as *latent space*. The parameters W of the flow are set in such a way that:

$$\mathbf{x} = \phi_W(\theta) \sim \mathcal{N}(\mathbf{x}|0,\mathbf{1}) \quad \text{if } \theta \sim p(\theta) \tag{16}$$

In other words, a normalizing flow defines a parametric representation of a generic probability distribution $p(\theta)$, obtained by change of variables

$$p_W^{\text{flow}}(\theta) = \mathcal{N}(\phi_W(\theta)|0,\mathbf{1})|\det J_{\phi_W}(\theta)| \tag{17}$$

where J_{ϕ_W} is the Jacobian of the flow transformation ϕ_W. Sampling from p_W^{flow} can then be easily done by sampling $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|0,\mathbf{1})$ and obtaining θ from the inverse flow transformation: $\theta = \phi_W^{-1}(\mathbf{x})$. Thus, given a target distribution, both the problems of sampling and of density estimation become tractable thanks to the normalizing flow model.

The flow transformation ϕ_W is built by *composing* n_{layers} simple (invertible) transformations, each called a layer. Of course, depending on the application, a variety of options are available in the literature. We build a layer by concatenating a linear transformation and a masked autoregressive layer [80–82] with n_{hidden} hidden features. A masked autoregressive layer implements the following transformation:

$$T_{\text{MADE}}(\theta) = a(\theta)\theta + b(\theta) \tag{18}$$

where the coefficients $a(\theta)$, $b(\theta)$ are computed by (masked) autoencoders with n_{hidden} hidden features.

In our case, the target probability distribution has support in the rectangle $[\theta_{\min}, \theta_{\max}]$, while the base distribution of the flow (a Gaussian) has support in $\mathbb{R}^D$. We implement the change of support explicitly by introducing the following transformation $T_0(\theta)\colon [\theta_{\min}, \theta_{\max}] \to \mathbb{R}^D$ as the first layer of the flow:

$$T_0(\theta) = 0.5 \log \frac{1+y}{1-y} \quad \text{with} \quad y = \frac{2\theta - \theta_{\min} - \theta_{\max}}{\theta_{\max} - \theta_{\min}} \tag{19}$$

where the fraction above is intended as element-wise division.[4] This transformation maps the rectangle $[\theta_{\min}, \theta_{\max}]$ into the plane. Then the remaining transformations only need to implement a change in probability density and not in the support of the distribution, making the loss function optimization easier.

The flow probability distribution $p_W^{\text{flow}}(\theta)$ is trained to closely reproduce a given probability distribution $p^{\text{target}}(\theta)$. During the training, the weights W of the flow are set by minimizing a loss function $\mathcal{L}_\phi(W)$, which measures the discrepancy between p^{target} and p_W^{flow}. The minimization is performed by gradient descent. In our case, $p^{\text{target}} \propto \sqrt{\det M}$, with an unset normalization.

Depending on the nature of the data, several loss functions are available. If *samples* from the target distribution are available, the loss function is defined as the forward Kullback–Leibler (KL) divergence between the target distribution $p^{\text{target}}(\theta)$ and the one defined by the flow in Eq. (17):

$$\mathcal{L}_\phi^{KL}(W) = -\mathbb{E}_{p^{\text{target}}(\theta)}[\log p_W^{\text{flow}}] + \text{const} \tag{20}$$

where the expected value is computed using empirical samples from $p^{\text{target}}(\theta)$ to provide a Monte-Carlo estimation of the loss function.

In our situation however, we do not have access to such samples (indeed, we are training the flow precisely to avoid sampling) but we are only able to evaluate p^{target} up to a constant scaling factor. For this reason, we treat the training as a *regression* problem, rather than a density estimation problem, and we use the following loss function:

[4] Note that the inverse T_0^{-1} of the transformation takes a simple form: $\frac{1}{2}[\tanh(T_0(\theta))(\theta_{\max} - \theta_{\min}) + \theta_{\max} + \theta_{\min}]$, where again the multiplication is intended as element-wise.

$$\mathcal{L}_\phi(W) = \frac{1}{N}\sum_{i=1}^{N}(\log p_W^{\mathrm{flow}}(\theta_i) - \log p^{\mathrm{target}}(\theta_i))^2$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left(\log p_W^{\mathrm{flow}}(\theta_i) - \log\sqrt{|M(\theta_i)|} + C\right)^2 \quad (21)$$

where the sum runs on a dataset of N points:

$$\{(\theta_i, \sqrt{|M(\theta_i)|})\}_{i=1}^{N} \quad (22)$$

Our experiments show that $N \simeq 5\times 10^5$ is adequate in most cases.

In Eq. (21), C is a *trainable* constant, which sets the normalization of $p^{\mathrm{target}} = e^{-C}\sqrt{|M|}$ on the domain of interest. Although not strictly needed, it can have a large impact on the flow performance, since it constrains the values of $\sqrt{|M(\theta)|}$ to a scale which is easier to learn by the normalizing flow. Some heuristics suggest initializing the constant to the 90th percentile of the values $\log\sqrt{|M(\theta)|}$ stored in the dataset. As shown in Appendix C, the constant can be used to compute (an approximation to) the volume of the parameter space $\mathcal{V}$ in Eq. (14).

The values of θ_i in Eq. (21) are obtained by sampling the masses m_1, m_2 from

$$p(\mathcal{M}_c, \eta) \propto \mathcal{M}_c^{10/3}\eta^{8/5} \quad (23)$$

where $\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1+m_2)^{1/5}}$ is the chirp mass and $\eta = \frac{m_1 m_2}{(m_1+m_2)^2}$ is the symmetric mass ratio. All other quantities are sampled from a uniform distribution in the coordinates.

Equation (23) defines a flat distribution on the chirptime parameters τ_0 and τ_3 [52]. Indeed, it can be shown that for a nonspinning binaries, the metric expressed in the chirptime coordinates is approximately flat [31,32], and that Eq. (23) represents a first order approximation to the true metric. Sampling from Eq. (23), ensures a high quality training set, where the distribution of the training points is reasonably close to the target distribution.[5]

During the training we halve the learning rate each time the validation loss does not improve more than a given threshold after a given number of iterations. This procedure finds local minima better in the loss function. We also apply early stopping, to avoid useless gradient descent iterations.

The training of the normalizing flow usually takes $\mathcal{O}(30$ minutes). On the other hand, from one to a few hours are needed to generate a dataset of $\mathcal{O}(10^5)$ points, depending on the dimensionality of the manifold and on the waveform approximant. This is the bulk of the cost of generating a template bank: the random template placing takes only a few minutes.

C. Random template placing

As customary, the input parameter controlling the average spacing and number of templates is the *minimal match* MM. It is defined as the minimum tolerable match that a random signal (inside the relevant parameter space) must have with its nearest templates in the bank. Of course, during the template placement, we only consider the match between templates on the same manifold, while the quantity can be used also to compare waveforms on different manifolds.

To generate our random template bank, following [45], we add random templates to the bank until a satisfactory coverage is achieved. The coverage is checked using a procedure that closely matches [50]. The templates are sampled from the normalizing flow in Eq. (17), which, as discussed above, is trained to target Eq. (15). This choice makes sure that the templates are spread as "uniformly as possible" across the manifold.

One point of the space θ is said to be *covered* by the bank if there is at least one template θ_T in the bank, whose squared metric distance (mismatch) as given in Eq. (11) is at most $1 - MM$ or:

$$d_M^2(\theta, \theta_T) < 1 - MM. \quad (24)$$

The covering fraction $\hat{\eta}$ of a given region $\mathcal{T}$ of the parameter space is then defined as the fraction of volume covered by the bank:

$$\hat{\eta}(\mathcal{T}) = \frac{1}{\mathrm{Vol}(\mathcal{T})}\int_{\mathcal{T}} d^D\theta\,\sqrt{\det M(\theta_i)}\,c(\theta) \quad (25)$$

where $c(\theta)$ is an indicator function:

$$c(\theta) = \begin{cases} 1 & \text{if } \theta \text{ is covered by the bank} \\ 0 & \text{otherwise.} \end{cases} \quad (26)$$

We do not require that the space is fully covered but we only require that it is covered with probability η. This means that we terminate the bank construction when the covering fraction $\hat{\eta} \geq \eta$.

To provide a sensible estimate of the covering fraction $\hat{\eta}$, we perform a Monte Carlo estimation of the integral in Eq. (25) [50]:

$$\hat{\eta}(\mathcal{T}) \simeq \frac{1}{N_{\mathrm{livepoints}}}\sum_i c(\theta_i) \quad (27)$$

where the $N_{\mathrm{livepoints}}$ samples $\theta_i \sim p^{\mathrm{flow}}$ are sampled from the normalizing flow and are called *livepoints*. Note that in Eq. (27), we do not compute volumes using the volume

[5] Indeed more samples are present at low chirp mass, which is where the metric determinant tends to have larger values due to longer waveforms (for a constant starting frequency). Hence, a consistent bias in the low mass region is largely penalized in the loss function due to more samples in the dataset at low mass.

element $\sqrt{\det M}$ itself but rather its normalizing flow approximation.

In practice, while the templates are being added to the bank, the distance between each livepoint is computed. If the ith livepoint is close enough to the newly added template, it will be removed from the set of livepoints and a running estimate of $\hat{\eta}(\mathcal{T})$ will be updated. The estimation of the covering fraction $\hat{\eta}$ has standard deviation ([50], Appendix A):

$$\sigma_{\hat{\eta}} = \sqrt{\frac{\eta(1-\eta)}{N_{\mathrm{livepoints}} - 1}} \tag{28}$$

which suggests using a large number of livepoints for better estimation. In [50], the authors typically choose $\eta = 0.9$ and $N_{\mathrm{livepoints}} = 2000$.

Since the method does not check for distances between templates, it can overcover the space (as also reported in [45,50]), especially for a low number of dimensions. Despite this, it is very fast and provides a reliable bank at a cheap computational and memory cost. Moreover, as argued in [45,53,54], for a large number of dimensions, the banks generated by the random method provide close to optimal performance.

As a final remark, we note that for the purpose of computing the covering fraction, the templates do not need to be stored, which enables the algorithm to run with a very low memory footprint. As exemplified in Sec. V C, this allows to study the number of templates required to cover a particular region of the parameter space, providing invaluable pieces of information useful to plan a GW search.

III. VALIDATION

In this section, we assess the performance of the two key ingredients of our template bank generation algorithm, namely the normalizing flow model and the random placement algorithm. Our goal is to understand the limitations of our algorithm as well as to make an informed choice of the various hyperparameters that impact the quality of the template bank.

We will consider different manifolds, which will be named with a string that lists the manifold coordinates. The coordinates are grouped by mass coordinates, spin coordinates, (eventual) eccentricity coordinates (i.e., e and a) and (eventual) angles coordinates (i.e., ι and φ). Consequently, a string has the format `Masses_Spin1_Spin2_Eccentricity_Angles`.

Valid options for the mass coordinates are `m1m2` which uses m_1 and m_2 as coordinates, `Mq` which uses total mass $M = m_1 + m_2$ and mass ratio $q = m_1/m_2 > 1$, and `logMq` which uses $\log_{10} M$ instead of M. Similarly, other variables are listed by their names. The manifold with spin label `chi` uses the effective spin parameter

$$\chi_{\mathrm{eff}} = \frac{m_1 s_{1z} + m_2 s_{2z}}{m_1 + m_2} \tag{29}$$

as coordinate. Since χ_{eff} is degenerate in the two spins, we choose to set $s_{1z} = s_{2z} = \chi_{\mathrm{eff}}$ and all the other spin components to 0.

If more than one spin coordinate is given for a given BH, the spin vector $\mathbf{s}$ will be parametrized in spherical coordinates with magnitude $s \in [0, 1)$ and angles $\theta \in [-\pi, \pi]$ and $\varphi \in [0, \pi]$ as follows:

$$s_{\mathrm{x}} = s \sin\theta \cos\phi \tag{30}$$

$$s_{\mathrm{y}} = s \sin\theta \sin\phi \tag{31}$$

$$s_{\mathrm{z}} = s \cos\theta. \tag{32}$$

Note that the angle θ controls the amount of precession. With $\theta = 0, \pm\pi$ the spin has only a z component (i.e., is aligned with the orbital angular momentum), while for $\theta = \pm\pi/2$ there is maximal precession, as the spin vector only has an in-plane component.

A. Normalizing flow validation

To study the accuracy of the normalizing flow model in reproducing $\sqrt{|M|}$, we consider five manifolds. The manifolds are listed in Table I, together with the region of the parameter space they cover. We also report the waveform approximant used as well as the frequency range where the metric is computed. The manifolds were chosen to have a variety of number of dimensions D and to cover a broad ranges of physical scenarios (nonspinning, aligned-spins, precession, HM, and eccentric orbits).

For each manifold we generate a dataset of 3×10^5 points and we compute the (log) value of the PDF in Eq. (15). We then train a normalizing flow model on each of the datasets. The architecture of each flow is also reported in Table I.

Figure 1 shows a histogram with the accuracy of the normalizing flow reconstruction of the PDF on each manifold. This is quantified by $\log_{10}\frac{p^{\mathrm{flow}}}{p^{\mathrm{true}}}$, which measures the logarithmic ratio between the two PDFs.

Overall, the accuracy of the flow is (almost) always contained within one order of magnitude. Whether a similar error is acceptable for the purpose of template placement needs to be checked on a case-by-case basis with an injection study, as discussed in Sec. IV.

We note that all histograms are well-centered around 0, showing that the flow does not have a systematic bias. Moreover, the accuracy tends to be higher for low-dimensional manifolds. Indeed, low dimensional manifolds present an easier learning task for the flow.

TABLE I. Details of the manifold considered for the validation of the normalizing flow model in Fig. 1. For each manifold, we report the variables being sampled together with their ranges. We also list the frequency range considered, the waveform approximant used, the number of dimensions D of the manifold as well as the number of hidden features for each layer of the flow.

	Parameter space	D	Architecture
m1m2_nonspinning	$m_1, m_2 \in [1, 200] M_\odot$ $q \in [1, 30]$ $f \in [15, 1024]$ Hz IMRPhenomD [83]	2	60 60 30
Mq_s1xz	$M \in [25, 100] M_\odot$ $q \in [1, 5]$ $s_1 \in [0, 0.99]$ $\theta_1 \in [0, \pi]$ $f \in [15, 1024]$ Hz IMRPhenomXP [84]	4	70 70
m1m2_chi_e	$m_1, m_2 \in [1, 50] M_\odot$ $q \in [1, 20]$ $\chi_{\mathrm{eff}} \in [-0.99, 0.99]$ $e \in [0, 0.5]$ $f \in [10, 1024]$ Hz EccentricFD [85]	4	60 60 60
logMq_s1z_s2z_iota (with HM)	$m_1, m_2 \in [50, 300] M_\odot$ $M \in [100, 400] M_\odot$ $q \in [1, 10]$ $s_{1z}, s_{2z} \in [-0.99, 0.99]$ $\iota \in [0, \pi]$ $f \in [10, 1024]$ Hz IMRPhenomXP [84]	5	20 60 60
logMq_s1xyz_s2z_iota	$m_1, m_2 \in [1, 100] M_\odot$ $M \in [2, 150] M_\odot$ $q \in [1, 20]$ $s_1 \in [0, 0.99]$ $\theta_1 \in [-\pi, \pi]$ $\phi_1 \in [0, \pi]$ $s_{2z} \in [-0.99, 0.99]$ $\iota \in [0, \pi]$ $f \in [15, 1024]$ Hz IMRPhenomXHM [86]	7	100 60 60 60

The manifold logMq_s1xyz_s2z_iota shows the largest spread in accuracy, as it is the largest dimensional manifold being considered. Note that it parameterizes a huge parameter space, which cannot be realistically covered by a template bank. Hence, as a realistic bank will necessarily cover a subset of the manifold, a flow trained on that smaller parameter space will most certainly show better accuracy, due to an easier regression task.

Finally, we see that the flow trained on the eccentric manifold m1m2_chi_e has remarkably good performance. This can be explained by the fact that the approximant EccentricFD [85] used is analytical. This ensures very smooth behavior across the parameter space, which can be easier for the normalizing flow model to learn.

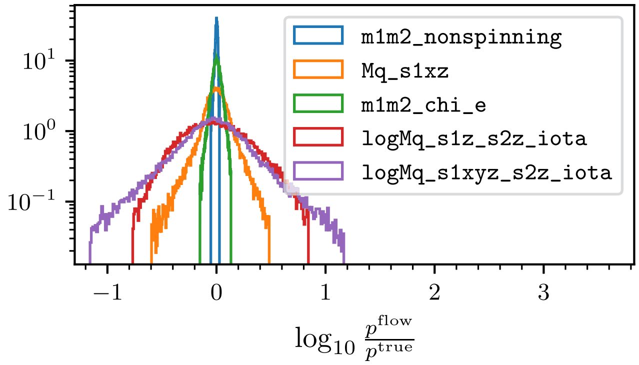

FIG. 1. Study of the accuracy for several normalizing flow, trained on different manifolds. For each manifold, we compute the logarithmic ration $\log_{10} \frac{p^{\mathrm{flow}}}{p^{\mathrm{true}}}$ between the PDF computed by the flow and the true one. We use 40000 test points from the validation set of each manifold. Details on the manifold considered are reported in Table I.

B. Template placement performance

As already stated, the template placement method in use closely matches the one introduced in [50]. The main novelty introduced here is sampling with the normalizing flow as opposed to rejection sampling.

For the random placement method, there are two parameters to tune that affect the final bank size. They are the number of livepoints $N_{\mathrm{livepoints}}$ and the covering fraction η. The authors of [50] make an extensive investigation on how the bank size depends on such quantities and we do not repeat such in-depth studies here.

We limit ourselves to examining the convergence of the template number $N_{\mathrm{templates}}$ as a function of $N_{\mathrm{livepoints}}$ (see [50], Fig. 4 (right)) in the case of manifolds with precessing and HM signals. For the study, we chose the manifolds m1m2_nonspinning, Mq_s1xz and logMq_s1z_s2z_iota introduced in Sec. III A (see also Table I). The second manifold covers a precessing parameter space, while the metric on the latter manifold is computed with an HM approximant [86].

We present our results in Fig. 2, where the number of templates is computed with a covering fraction $\eta = 0.9$ with varying $N_{\mathrm{livepoints}}$. In all cases the number of templates converges to a constant value as $N_{\mathrm{livepoints}}$ increases. Already ~ 500 livepoints are enough to provide an accurate estimation of the bank size. Our results are consistent with the findings of [50], which we further extend to higher-dimensional manifolds.

IV. COMPARISON WITH OTHER BANK GENERATION METHODS

We compare the output of MBANK with two banks available in the literature, generated with two different methods. The first bank is a nonspinning HM bank [21], covering the high mass region of the BBH parameter space.

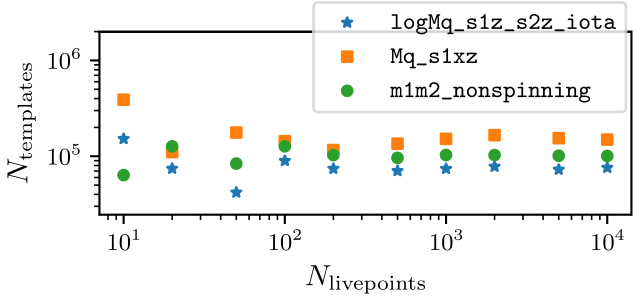

FIG. 2. Validation of the random template placement algorithm. For three of the manifolds introduced in Table I, we plot the number the number of templates $N_{\text{templates}}$ of a random template bank as a function of the number of livepoints $N_{\text{livepoints}}$ used to estimate the covering fraction. For each template bank, we set $\eta = 0.9$ and $MM = 0.97$.

The bank was generated using the stochastic placement algorithm, as implemented in the code SBANK [40]. The second bank is the aligned-spin bank [56] currently in use by the GstLAL pipeline [64,66] for the fourth observing run (O4) of the LIGO-Virgo-Kagra collaboration. It was generated using the MANIFOLD [51] metric template placement algorithm called and covers a very wide mass range in the BNS and BBH parameter space. Both banks have a minimal match MM requirement of 0.97.

In much of what follows we will measure the coverage of a bank. To do so, we randomly extract a number of simulated signals and, for each of them, we compute the maximum match with the templates of the bank. The latter quantity is called *fitting factor* FF which, for a simulated signal characterized by orbital parameters θ, it is defined as:

$$FF(\theta) = \max_{\theta' \in \text{bank}} \mathcal{M}(\theta, \theta') \tag{33}$$

Clearly, the match is computed using Eq. (6).

Borrowing the jargon of GW searches, we call *injections* the simulations for which we evaluate the fitting factor. In a real search, such signals would be added to the interferometer's data (i.e., injected) to measure the performance of the pipeline: the fitting factor measures the fraction of SNR lost due to the discreteness of the template bank.

A. A nonspinning HM template bank

The nonspinning HM bank described in [21] covers systems with total mass M in the range $[50, 400] M_\odot$ and mass ratio $q \in [1, 10]$. It also includes the inclination angle ι and reference phase φ of the system, both covering the whole possible spectrum of values $\iota \in [0, \pi]$ and $\varphi \in [0, 2\pi]$. The authors use the analytical "zero-detuning high power" PSD [87] and consider a low frequency cutoff $f_{\text{min}} = 10$ Hz.

As already noted, they use the state-of-the-art code SBANK [39,40]. The method is very accurate and known to provide effective coverage with a low number of templates. Of course, this comes at a large up-front computational cost to construct the bank.

TABLE II. Details of the two banks available in the literature that we reproduce with our code. For each bank, we indicate the parameter space considered and the approximant used. We also compare the number of templates of the banks obtained with the different methods.

		Size	
	Parameter space	Original	MBANK
HM bank [21]	$M \in [50, 400] M_\odot$ $q \in [1, 10]$ $\iota \in [0, \pi]$ $\varphi \in [0, 2\pi]$ IMRPhenomXHM [86]	20500	58932
"All-sky" bank [56]	$m_1, m_2 \in [1, 200] M_\odot$ $q \in [1, 20]$ $\chi_{\text{eff}} \in [-0.99, 0.99]$ IMRPhenomD [83]	1.8×10^6	1.3×10^6

To reproduce this bank, we place templates on the manifold logMq_nonspinning_iotaphi, with coordinates $\log_{10} M$, q, ι and φ. We use the same PSD and coordinate ranges as the original bank. We refer to our bank as "HM bank." We train a normalizing flow model with 4 layers with 60, 60, 60, 10 hidden features respectively and we choose $N_{\text{livepoints}} = 2000$ and a covering fraction $\eta = 0.8$. Our bank has 58932 templates and took a few hours to generate; the original bank is reported to have 20500 templates. All information is summarized in Table II. We perform an injection study, drawing 10^5 signals uniformly sampled in $\log M$, q, $\cos \iota$ and φ. The results of such study are reported in Figs. 3 and 4.

First we note that our bank successfully covers the parameter space, with only 1% of injections found with fitting factor below 0.97 and less than 1% with fitting factor

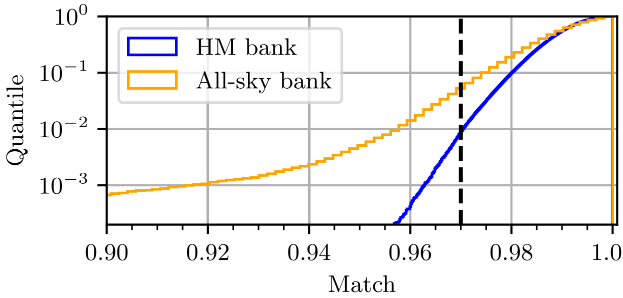

FIG. 3. Fitting factor studies for the two template banks introduced in Sec. IV. As discussed in Sec. IVA and IVB respectively, "HM bank" is designed to reproduce [21] and targets high mass non-spinning systems with HM content, while the "All-sky bank" bank covers aligned-spin systems (without HM) over a broad mass range, following [56]. We report the cumulative histogram of the fitting factors of 10^5 injections samples across the parameter space.

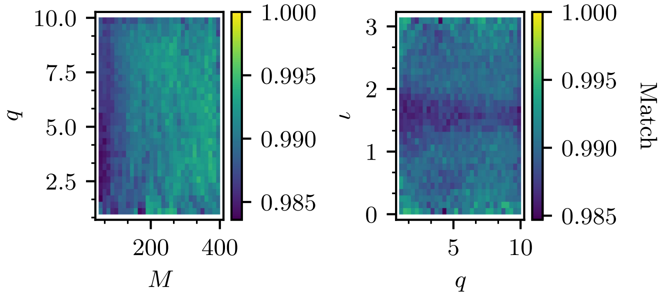

FIG. 4. Validation of the “HM bank,” generated with our code and designed to reproduced [21]. For each two dimensional bin, we report the median fitting factor of 10^5 injections covering the parameter space, as described in the text.

below 0.96. The coverage of the bank is similar to that of [21]. In Fig. 4, we observe that the coverage is uniform across the space, i.e., we do not see regions where the fitting factor is significantly different from the others.

Comparing the number of templates, it is striking that our bank has almost three times more templates than the original template bank. As no template rejection is done during the random bank construction, there is no control over templates being too close to each other. For this reason, an over-coverage of the space is inherent to the random template placement and is also reported in [45,50]. This problem can be addressed in future work, as discussed in Sec. VI.

B. An “all-sky” template bank

The aligned-spin bank (with no HMs) introduced in [56] covers a broad mass range, with systems with component masses $m_1, m_2 \in [1, 200]M_\odot$. The spins of the two objects are constrained to be equal to each other,[6] $s_{1z} = s_{2z} = \chi_{\mathrm{eff}}$, spanning the range $[-0.99, 0.99]$. The authors set an upper limit to the mass ratio $q < 20$. Moreover, for objects with component mass $m < 3M_\odot$, they limit χ_{eff} in the range $[-0.05, 0.05]$.[7] The authors use the Advanced LIGO O4 Design PSD (with 190 Mpc range) [89] and consider a low frequency cutoff $f_{\mathrm{min}} = 10$ Hz.

The comparison with [56] is particularly interesting, since the bank is also produced with a metric template placement, implemented in the MANIFOLD code [51]. MANIFOLD uses a geometric approach, where the parameter space is iteratively split into (hyper)rectangles along the coordinates, until the volume of each rectangle reaches a sufficiently small value that it can be covered by a single template.

As summarized in Table II, we construct a bank to cover the parameter space used in [56] over the manifold `m1m2_chi`, sampling the coordinates m_1, m_2, and χ_{eff}. We may refer to our bank as the “all-sky” bank. To produce our “all-sky” bank, we trained three different normalizing flows in different regions of the parameter space. A first normalizing flow covers the BBH region with $m_1 \in [3, 200]M_\odot$, with $\chi_{\mathrm{eff}} \in [-0.99, 0.99]$. A second one covers the BNS region, covering the manifold, $(m_1, m_2, \chi_{\mathrm{eff}}) \in [1, 3]M_\odot \times [1, 3]M_\odot \times [-0.05, 0.05]$. A third normalizing flow specializes in the high mass region, characterized by $m_1, m_2 \in [100, 200]M_\odot$. Indeed, at high masses, the template density is so low that hardly any livepoint is sampled, which results in dramatic under-coverage. An appropriate coverage is enforced by the third normalizing flow, which places $\mathcal{O}(3000)$ templates in the region as opposed to *zero* templates placed by the first flow. The additional coverage at high masses is manifest in Fig. 5, as discontinuity in the fitting factor for $m_1, m_2 > 100M_\odot$.

All the three normalizing flow models are made of 5 layers of 10 hidden features each. Three templates banks are generated using each normalizing flow and they are merged together afterward. For the template placement we set $N_{\mathrm{livepoints}} = 2000$ and covering fraction $\eta = 0.95$. The resulting bank has 1326805 templates.

The bank generation took around three hours, with most of the computing time spent on the dataset generation (i.e., on expensive metric computation). If needed, the dataset generation can be easily parallelized using MBANK, hence reducing significantly the bank generation time. Relying on parallel execution, [56] reported a generation time of minutes.

To validate our bank, we generate an injection set with 10^5 injections, with the logarithm of the masses uniformly sampled. Results of our injections studies are reported in Figs. 3 and 5. Note that our injection set is different from the ones used in [56].

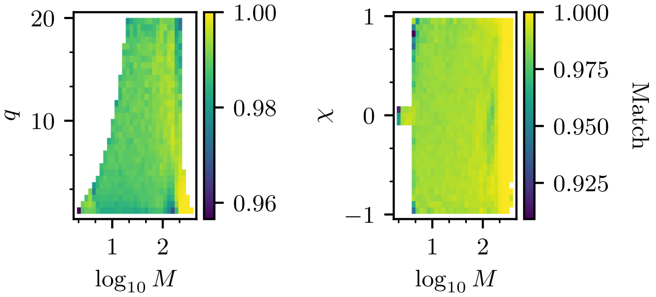

FIG. 5. Validation of the “All-sky bank,” generated with our code and designed to reproduced [56]. For each two dimensional bin, we report the median fitting factor of 10^5 injections covering the parameter space, as described in the text.

[6] This choice reduces the dimension of the manifold, without compromising the template bank accuracy.

[7] This is motivated by astrophysical considerations. Objects with masses smaller than $3M_\odot$ are likely to be neutron stars and such objects are believed to develop only mild rotations [88].

In Fig. 3, we see that ~5% of the injections have a match below 0.97. The low fitting factor injections are mostly located around the low mass corners of the bank, clustered on the low mass end of the BNS region and in the high spin—low mass edge of the BBH region. Inside the template bank and on the high mass end of the parameter space, satisfactory coverage is achieved. Our results suggest that MBANK struggles to accurately cover the "narrow" corners of the parameter space. Nevertheless, this is a common problem that has been observed with other placement methods as well, and several strategies have been proposed to cope with it. Within our framework, the simplest option would be to extend the boundaries of the bank at low masses, thus ensuring better coverage of the region of interest.

With slight variations depending on the region of parameter space, [56] reports that 10% of BBH injections have fitting factor smaller than ~0.98, while for our bank the 10th percentile is around 0.975. Even though it is hard to compare the results directly due to different injection sets, it seems fair to state that, compared to [56], our template bank provides slightly worse injection recovery. On the other hand, our template bank has 30% *less* templates, matching the number of templates placed by SBANK in the same region, as reported by [56]. With an accurate treatment of the low mass corner, the coverage of our template bank will easily match the one of [56], with a comparable bank's size.

V. NOVEL APPLICATIONS OF THE METHOD

Our template placement method allows for several exciting applications in GW data analysis. Obviously, the most straightforward application is the generation of high-dimensional template banks, such as a precessing and/or HM banks. While in principle it is possible to generate these high-dimensional banks with a stochastic placement method, very few of such banks have been generated so far, mostly due to the enormous computational cost of choosing the right parameter space and of computing the match between templates. Their generation becomes feasible thanks to MBANK.

Besides efficient high-dimensional bank generation, our method can be used for other purposes as well. These include choosing the appropriate parameter space to cover by forecasting the size of a bank or selecting the appropriate coordinates to cover a given region of binary systems. Moreover, our normalizing flow could be used as a proposal for a stochastic placement algorithm or to create datasets for machine-learning applications in GW data analysis.

In what follows, we generate a large precessing template bank and a large aligned-spin HM bank. Additionally, we provide a detailed discussion of other innovative applications of our code.

A. A precessing bank

1. Choosing the parameter space

The main difficulty in generating a precessing bank lies in the huge size of the parameter space. As we show below, a precessing bank can easily have *billions* of templates, even when covering the mass range routinely explored by "standard" searches. As current search pipelines can handle only up to a few *million* templates, due to computational cost limitations, the size of a bank sets very stringent constraints in the selection of a suitable parameter space to explore with a GW search.

Another difficulty, related to the first, arises from the choice of the BBH coordinates to include in the bank, i.e., the choice of manifold. In principle, a precessing BBH system is described by 10 parameters (two masses, six spins, and two angles). However, not all of them are important, as large changes in some parameters do not result in large changes in the waveform morphology. Thus, including them in the bank does not yield any obvious improvement and, on the contrary, it may lead to vanishing metric eigenvalues, which would degrade the metric predictivity, hence the template placement. The latter point is discussed with more details in Sec. V C.

Finally, a more technical complexity arises from the fact that in high dimensional spaces, both the training of a normalizing flow (see Sec. III A) and the template placement become harder, hence possibly harming the quality of the template bank.

All these difficulties imply that great care must be taken when deciding both the parameter space and the BBH variables to include in the bank. The choices are entangled, since covering different manifolds with the same mass range can produce banks of very different sizes. Roughly speaking, choosing a lower dimensional submanifold reduces the bank size, at the cost of a loss in the bank's ability to cover the high dimensional space.

To choose a manifold, we rely on the theory. In [70], the authors find that the effect of the four in-plane spin components (i.e., $s_{1\mathrm{x}}, s_{1\mathrm{y}}, s_{2\mathrm{x}}, s_{2\mathrm{y}}$) can be well approximated by a single precessing spin parameter χ_P assigned to the x-component of the heavier object's spin. Thus, a generic precessing system is roughly equivalent to a system with

$$\mathbf{s}_1 = (\chi_P, 0, s_{1\mathrm{z}})$$
$$\mathbf{s}_2 = (0, 0, s_{2\mathrm{z}})$$

effectively creating an explicit mapping between a six dimensional spin manifold to a three dimensional one. In a later work [90], it is suggested that to capture the combined effect of precession and HM, a two-dimensional spin parameter $\vec{\chi_P}$ is needed. In this case, the mapping is between a six-dimensional spin manifold to a four-dimensional one.

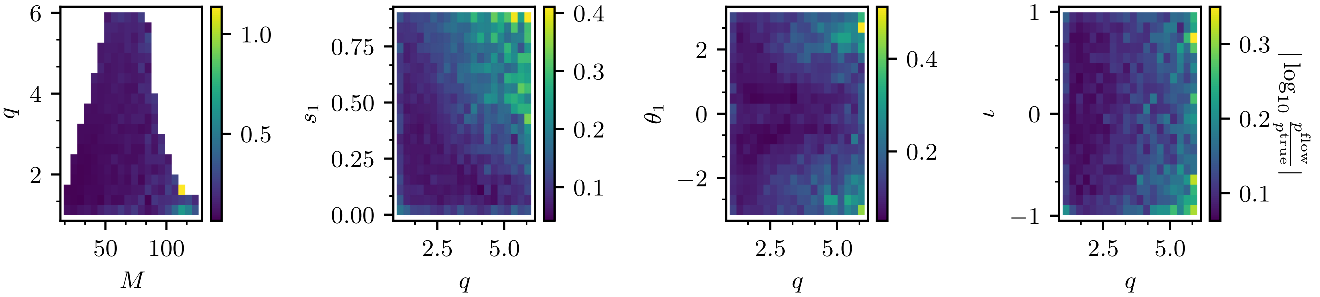

FIG. 6. Accuracy of the normalizing flow trained used to generate the precessing bank in Sec. V A. The accuracy is expressed in terms of the logarithmic ratio between the template density PDF p^{true} Eq. (15) and its approximation p^{flow} given by the flow. The flow accuracy is evaluated on 40000 test points.

Both works suggest that the in-plane components of the spin on the lighter object (i.e., $s_{2\mathrm{x}}, s_{2\mathrm{y}}$) can be neglected, reducing the dimensionality of the parameter space. Moreover, since we are not currently concerned with precession combined with HM,[8] we can rely on the one-dimensional effective spin mapping [70] to also neglect the y-component of the spin of the heavier object, $s_{1\mathrm{y}}$.

We then consider only three out of six spin components, $s_{1\mathrm{x}}, s_{1\mathrm{z}}$, and $s_{2\mathrm{z}}$, where all the effects of precession are included in $s_{1\mathrm{x}}$. To obtain accurate coverage, we also need to include the inclination ι in the manifold. Some investigations showed that the inclusion of the reference phase φ yields a (almost) degenerate metric, which, by dramatically undercovering the space, negatively affects the placement. Luckily, as injection studies show that neglecting φ does not harm the bank's effectiveness, we can exclude φ from the set of parameters. However, this might not be the case if we include both precession and HMs.

To summarize, we find that the 6 variables M, q, $s_{1\mathrm{x}}$, $s_{1\mathrm{z}}$, $s_{2\mathrm{z}}$, and ι provide a sufficiently complete description of waveforms in the precessing space. This claim is confirmed by an injection study presented in Fig. 9, where we see that more than 93% of the injections covering the 10 dimensional precessing space have a fitting factor greater than the minimal match target of 0.97. We note that a precessing template bank with HMs will likely need to sample two additional variables $s_{1\mathrm{y}}$ and φ, hence increasing the dimensionality to 8 [90].

Regarding the search parameter space, we are interested to target BBHs where precession is stronger as such systems are most likely to be missed by current searches [22,25]. Precession is more visible for high mass ratio, edge-on[9] systems and for high values of spins [91]. Moreover, as more cycles are detectable, precession effects will be stronger for longer signals due to the accumulation of the phasing effects of precession. These considerations suggest that very asymmetric, low mass systems, such as the neutron star-black hole (NSBH) space, would be an ideal target for a precessing bank. However, as shown below in Sec. V C searching the full NSBH region is unfeasible, as hundreds of millions of templates would be needed.

For this reason, we restrict ourselves to a different, less extreme, region of the parameter space. After several investigations, made possible by the speed and flexibility of our approach, we found that a parameter space with component masses in the range $[8, 70]M_\odot$, with a mass ratio cutoff of 6, produces a bank with a manageable size. In this space, we obtain a precessing bank with ~2millions templates. Extending the parameter space to lower masses (or higher mass ratios) results in much larger banks, pushing the limits of current pipelines.

In closing, we stress again that the investigations above are made possible by MBANK, since they rely on fast template bank generation across a variety of manifolds and ranges of coordinates.

2. Generating and validating the bank

As stated above, our precessing bank covers the manifold `logMq_s1xz_s2z_iota`, with coordinates $\log_{10} M$, q, s_1, θ_1, $s_{2\mathrm{z}}$, and ι. We consider BBHs with individual masses between 8 and $70M_\odot$, with a maximum mass ratio $q = 6$. The other variables s_1, θ_1, $s_{2\mathrm{z}}$, and ι cover the set $[0, 0.9] \times [-\pi, \pi] \times [-0.99, 0.99] \times [0, \pi]$.

To compute the metric, we use the Advanced LIGO O4 sensitivity estimate [89] and we set a frequency range of $[15, 1024]$ Hz, employing the approximant `IMRPhenomXP` [84]. We train a normalizing flow with 3 layers with 100, 100, and 60 hidden features respectively, using a dataset of 4×10^5 points. The flow performance after training is reported in Fig. 6. To generate the bank, we use a minimal match requirement of 0.97, with a covering fraction $\eta = 0.95$, estimated with 3000 livepoints. In a similar way to what was done for the "all-sky" template bank, we also train a normalizing flow to target the high total mass region with $M > 100M_\odot$. We use the latter to place templates with the same covering fraction $\eta = 0.95$, with great benefits. The overall bank has 1605625 templates, plotted in Fig. 16.

[8] In such a space, the template banks would be unfeasibly large.
[9] An edge-on system is observed with inclination $\iota \simeq \pi/2$.

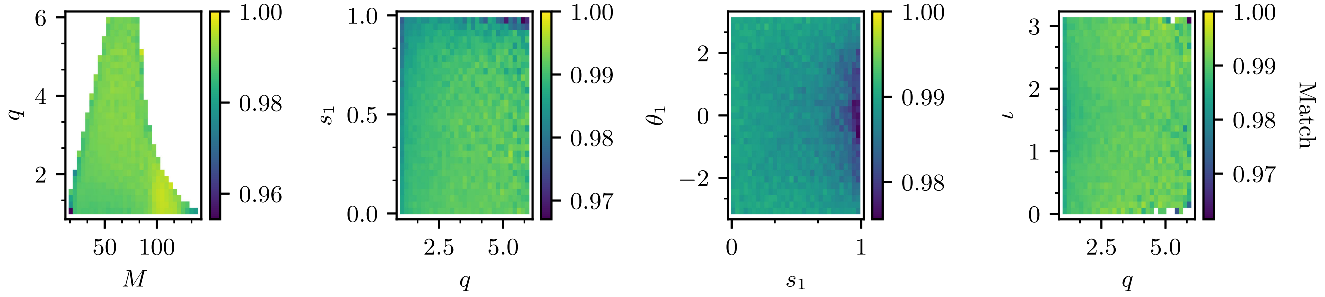

FIG. 7. Fitting factor study of the precessing bank, introduced in Sec. V A. For each bin, we color-code the median fitting factor of 10^5 injections sampled "on manifold," as described in the text.

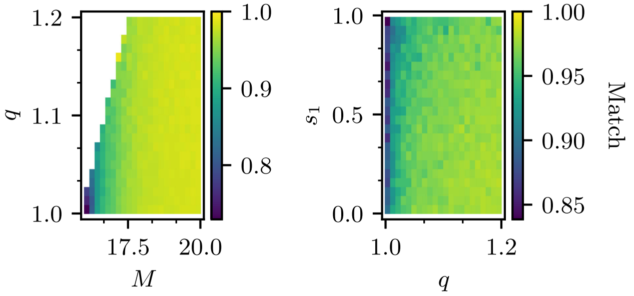

FIG. 8. Fitting factor study of the precessing bank of Sec. V A. Unlike Fig. 7, here we focus on the low q, low M region, where the random placement method fails. For each bin, we color-code the median fitting factor of 5×10^4 injections sampled "on manifold," as described in the text.

This bank generation took a few hours in total: ~1 hour for the dataset generation, ~30 minutes for the training of the flow and ~5 minutes for the template placing. All the steps above ran on a single core, using less than 4 GB of memory. We highlight that our time and memory requirements are a fraction of those of a similar bank with the state-of-the-art stochastic algorithm.

The template distribution reported in Fig. 16 shows a spike in the template density for $\theta_1 = \pm\pi$ (close to the nonprecessing limit) in the high mass ratio and high s_1 region. Some investigations indicate that these are not artefacts introduced by the normalizing flow. Whether the feature is physical or is due to the behavior of the waveform approximant in the nonprecessing limit remains an open question which needs more inspection.

To study the performance of our template bank, we generate two injections sets, with masses sampled uniformly in $\log m_1$ and $\log m_2$. The first set, labeled "full precessing" has fully precessing injections (with two $3D$ spins and varying φ). The second one, denoted as "on manifold," has injections lying on the manifold `logMq_s1xz_s2z_iota`, hence covering a subset of the "full precessing" set. The latter set is needed to asses the coverage of the bank on the manifold on which the templates lie and thus is a measure of the templates' placement accuracy. On the other hand, the "full precessing" injection set evaluates the ability of the bank to recover a generic precessing signal, hence assessing the quality of our choice of manifold. Clearly, this is the injection set that is most relevant for designing the bank for a fully precessing search.

We report the results of our study in Fig. 9, in the form of a histogram of the fitting factors, and in Fig. 7, where we study the dependency of the fitting factor across the parameter space. Figure 8 reports the same fitting factor study focused on the low q, low M region.

As is clear from Figs. 8 and 9, the random template placement method *fails* for the low q, low M region, with $q \leq 1.2$ and $M \leq 20M_\odot$, where only ~40% of the injections "on manifold" have a fitting factor higher than 0.97. On the other hand, outside the low q, low M corner, the template bank provides a good coverage: 97% of the injections "on manifold" has a fitting factor large than 0.97.

The poor performance for low mass ratio and low masses was also observed in the "all-sky" template bank in Sec. IV B, although less severe. Such failure be explained by two combined causes. First of all, as noted above, the random method is unable to cover "sharp" corners of the parameter space, due to the lack of appropriate boundary treatment: this can (and does) severely limit the bank's ability to cover the space. Moreover, we observe that for $q \to 1$ the metric determinant goes rapidly to 0, meaning that very few templates are placed. This is shown in Fig. 10, where we plot $|M|$ as a function of q keeping constant all the other coordinates.[10] The two effects combines together in the low q, low M region, which is drastically undercovered. The same issue is not observed anywhere else in the parameter space.

In principle, we could remedy the problem by extending the covered region to lower masses and higher q: this would make sure that the low q, low M target region does not lie at the boundaries of the bank anymore. However, the lack of coverage in this region is not a major concern for the bank's effectiveness in a real search scenario. Indeed, precession

[10] Although not reported here, the same behavior is observed for "standard" signals.

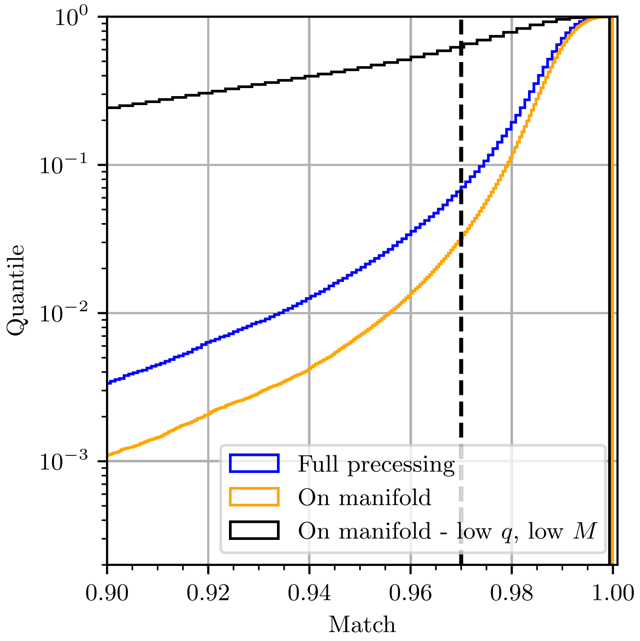

FIG. 9. Cumulative fitting factor for the precessing bank introduced in Sec. V A. The 10^5 injections "full precessing" have isotropic spins, while the 3×10^5 precessing injections "on manifold" are sampled on the manifold `logMq_s1xz_s2z_iota` and they have $s_{1y} = s_{2x} = s_{2y} = \varphi = 0$. For the injections "on manifold," we plot separately the low q, low M corner, characterized by $q \leq 1.2$ and $M \leq 20 M_{\odot}$. The other two histograms exclude this region.

for $q \sim 1$ has very little effect on the BBH waveform and a precessing system with symmetric masses would likely be detected by current aligned-spin searches.

In Fig. 7, we see that the coverage is rather uniform across the parameter space. The median fitting factor slightly drops for the high q high s_1 corner of the parameter space. As shown in Fig. 6, the flow performance degrades in that undercovered corner of the space: the true template density $\sqrt{|M|}$ is underestimated by the normalizing flow, which accordingly places less templates than optimal.

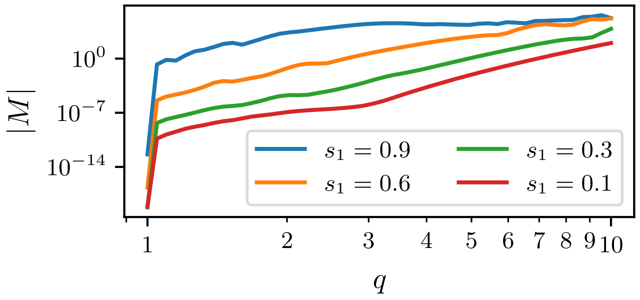

FIG. 10. Determinant of the metric $|M|$ as a function of mass ration q for different values of s_1. The metric is evaluated on the manifold `Mq_s1xz_s2z_iota`, with $M = 10 M_{\odot}$, $\theta_1 = \pi/2$, $s_{2z} = -0.3$, and $\iota = \pi/2$. It is manifest that in all cases, the metric determinant vanishes while $q \to 1$.

The fitting factor of the "full precessing" injection set is fairly good, with only 7% of the injections (outside the "low q, low M" region) below the target match. This means that the χ_P approximation that motivates our choice is robust: the manifold `logMq_s1xz_s2z_iota` provides a faithful low-dimensionality representation of the entire precessing parameter space.

B. An aligned-spin HM bank

In a sense, aligned-spin HM template banks are easier to generate than precessing ones, due to a smaller dimensionality of the parameter space. Indeed, a generic aligned-spin binary system with HMs is characterized by 6 parameters (two masses, two spins and two angles $\iota \varphi$) but, as for the non-HM case, the spin effects can be easily parametrized with an effective spin parameter, reducing the number of dimensions to 5. Note that here we deal with one dimension more than in the nonspinning HM bank produced in Sec. IV A. Despite less uncertainties in the choice of manifold than in the precessing case, the parameter space is very large and producing a template bank of a feasible size still requires a careful choice of the region to target.

We used MBANK to generate an HM aligned-spin bank, covering the high mass region of the BBH parameter space. High mass events are notoriously hard to detect [92,93]. As they are very short, their morphology matches closely non-Gaussian transient noise bursts, also called *glitches*, [94–97]. In this scenario, a more realistic model for the waveform can improve the detectability of such signals, thanks to both an increase in recovered SNR and to a more accurate signal-based veto [64,98]. Several studies [27,69,99,100] confirmed this claim, finding that failing to consider HMs in GW searches can lead to a large sensitivity loss for large mass ratios $q \gtrsim 4$ and high masses $M \gtrsim 100 M_{\odot}$ [91].

Consequently, our bank covers the manifold `logMq_chi_iotaphi`, sampling $\log_{10} M$, q and χ_{eff} as well as inclination and reference phase. We consider templates with total mass M between $50 M_{\odot}$ and $400 M_{\odot}$ and a mass ratio smaller than 7. The effective spin lies in range $[-0.99, 0.99]$ and, as usual, $\iota \in [0, \pi]$ and $\varphi \in [-\pi, \pi]$. We use the Advanced LIGO O4 sensitivity estimate [89] and we set a frequency range of $[10, 1024]$ Hz, with approximant `IMRPhenomXHM` [86].

We generate a dataset with 4×10^5 points and train a normalizing flow with 4 layers, each with $n_{\text{hidden}} = 60$ hidden features. The accuracy of the normalizing flow is reported in Fig. 11. For the template placement, we use a minimal match requirement of 0.97 and set a covering fraction $\eta = 0.8$, estimated with 10000 livepoints. The overall bank gathers 2115299 templates, which are plotted in Fig. 17. The bank generation took roughly the same time as for the precessing bank.

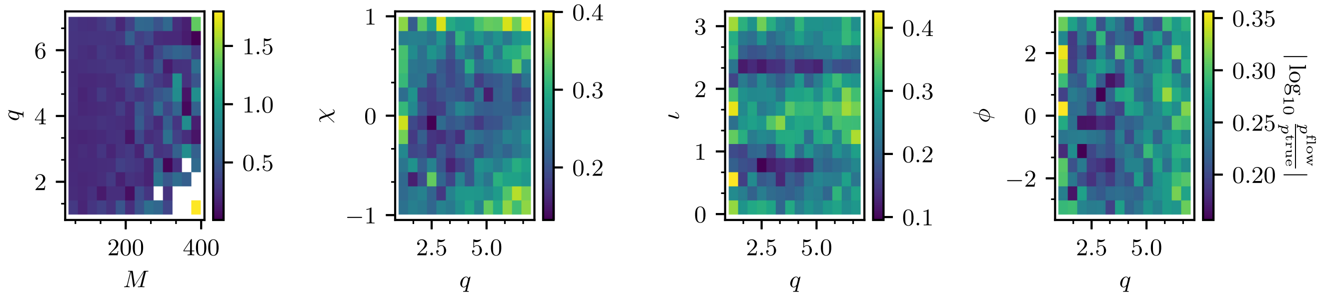

FIG. 11. Accuracy of the normalizing flow trained used to generate the aligned-spin HM bank of Sec. V B. The accuracy is expressed in terms of the logarithmic ratio between the template density PDF p^{true} Eq. (15) and its approximation p^{flow} given by the flow. The flow accuracy is evaluated on 40000 test points.

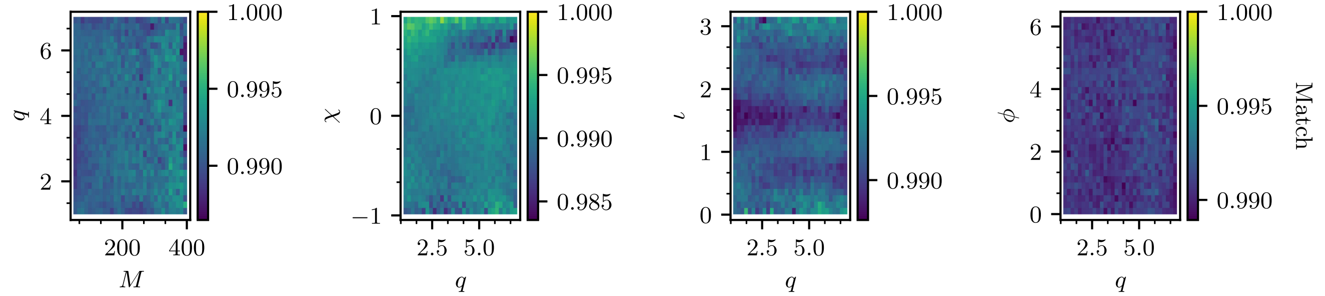

FIG. 12. Fitting factor study of the aligned-spin HM bank, introduced in Sec. V B. For each bin, we color-code the median fitting factor of 10^5 injections sampled uniformly from the parameter space.

We study the bank performance with 10^5 injections and report their fitting factor in Figs. 12 and 13. Our injection study shows that only ∼2% of the injections have a fitting factor smaller than the target of 0.97, with a median fitting factor of 0.99. We can conclude that the bank provides good coverage of the parameter space. Moreover, the fitting factor is rather constant across all the parameters space. As was also the case for the HM bank introduced in Sec. IV A, there are not regions which are undercovered by the template banks. Also the accuracy of the normalizing flow does not vary too much over the parameter space, showing a bad performance only in the region with high total mass and low mass ratio.

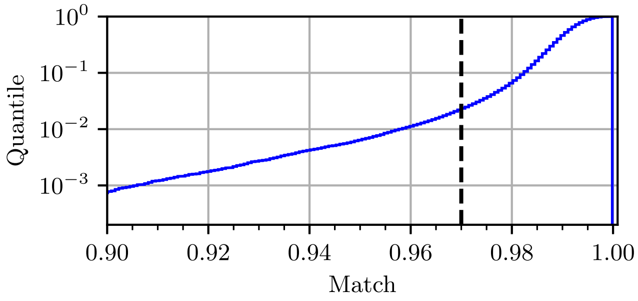

FIG. 13. Cumulative fitting factor for the aligned-spin HM bank described in Sec. V B. The histogram is built upon 10^5 injections sampled from the manifold.

We note that, in order to achieve good performance in the two HM banks presented in this work, we set a covering fraction of only $\eta = 0.8$. This is significantly lower than what we used for the non-HM banks and also lower than the recommended value of $\eta = 0.9$ in [50]. This means that, unlike the non-HM case, the metric match in Eq. (11) *underestimates* the “true” match. In this scenario, the covering fraction estimated with the livepoints (which makes use of the metric) also underestimates the “true” covering fraction. Therefore, a lower value of η is enough to obtain an acceptable coverage. This is not the case for non-HM banks. The reason why this happens only for HM banks is currently not understood and requires more investigation.

C. Other possible applications

The speed of the bank generation, together with the flexibility of the flow in sampling from the parameter space, allows for several novel applications of our work to GW data analysis, besides producing high-dimensional template banks. Without being exhaustive, we discuss below some of the new possibilities.

a. Selecting the parameter space to target As already discussed, the choice of the parameter space to target in GW searches can be challenging, as it is hard to obtain a reliable forcast of the number of templates needed for accurate coverage. Moving toward high-dimensional template banks, the number of templates increases by orders of magnitude and the standard stochastic approach suffers from memory issues due to the storage of the waveforms needed for the match calculation. This in turn makes it difficult to even explore high-dimensional spaces, as the current algorithms time-out by the time the bank reaches several million templates.

Our method has a low memory footprint and this makes possible to forecast the number of templates in a given parameter space, providing invaluable information to choose an appropriate target for the search. To do so, the interested user might train a normalizing flow on a large region of the parameter space and then place templates in a subregion, without the need to store them. Sampling in a subregion can be easily completed with the use of rejection sampling.

A natural candidate to demonstrate the usefulness of this technique is the precessing NSBH parameter space. Indeed, due to the large mass asymmetry of NSBH systems (i.e., high q), precession has a strong imprint on the waveform, leading to a very large volume to cover by a template bank. To study the number of templates needed to cover the space, we train a normalizing flow model on the manifold `logMq_s1xz_iota`[11] for systems with masses $m_1 \in [10, 60] M_\odot$ and $m_2 \in [1, 3] M_\odot$, with mass ratios $q \in [3.3, 15]$. The other coordinates s_1, θ_1 and ι vary in set $[0, 0.9] \times [-\pi, \pi] \times [0, \pi]$. As above, we use the approximant `IMRPhenomXP`, in a frequency range of [15,1024] Hz.

To study the parameter space size, we run our template placement algorithm for varying maximum total mass $M_{\rm max}$ and we measure the number of templates needed to achieve a covering fraction of $\eta = 0.9$ for different minimal match requirements. Since we do not store and validate the template banks, there is no guarantee that the resulting banks provide a satisfactory coverage. The procedure is just meant to obtain an order of magnitude estimation of the bank size.

As shown in Fig. 14, the precessing NSBH parameter space is huge. With a minimal match requirement of 0.9, around 100 million templates are needed to cover the full space. Around half of the templates are in the low total mass region with $M \in [11, 15] M_\odot$. The numbers agree with the investigations carried out in [26]. To cover the space with a minimal match of 0.95, around five times more templates are needed.

The magnitude of the precessing NSBH space makes it nearly impossible to use traditional matched filtering techniques to search for such signals. It thus becomes compelling to either develop new search techniques [26] or to improve the computational power available.

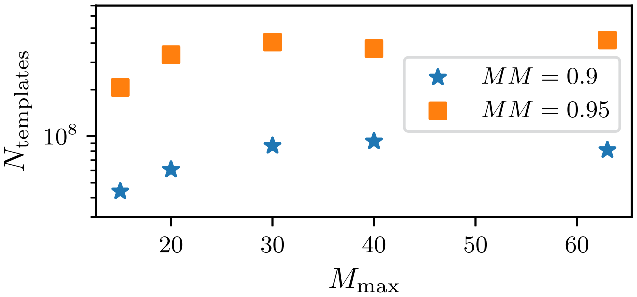

FIG. 14. Study of the size of a template bank in the neutron star-black hole parameter space. Each point refers to a template bank on the manifold `logMq_s1xz_iota`, covering a total mass range $M \in [M_{\rm min}, M_{\rm max}]$. The component masses are limited to $m_1 \in [10, 60] M_\odot$ and $m_2 \in [1, 3] M_\odot$, with mass ratios $q \in [3.3, 15]$. In the plot we report the number of templates $N_{\rm templates}$ as a function of the maximum total mass $M_{\rm max}$, for different minimal match requirements. The resulting banks are huge, with tens of millions of templates, showing that a search for precessing NSBH binaries is still prohibitively costly.

Thanks to our method, similar estimates can easily be done for other regions of the BBH parameter space (e.g. targeting eccentric BBHs), thus providing invaluable information to plan future high-dimensional GW searches.

b. Manifold selection The metric eigenvalues and eigenvectors can give an interesting piece of information about the relative importance of the coordinates of the manifold. Let λ_i and $\mathbf{v}_i$ be the ith eigenvalue and eigenvector respectively of the metric M_{ij}. We can think of each eigenvector λ_i as a measure of the relative importance of the eigenvector $\mathbf{v}_i$, which represents a linear combination of the coordinates. We can then introduce the following quantity for each coordinate j, which we call *coordinate importance*:

$$\mathcal{I}_j = \left| \sum_i \lambda_i (\mathbf{v}_i)_j \right| \tag{34}$$

where $(\mathbf{v}_i)_j$ is the jth component of the ith eigenvector. It is a weighted average over the projection of each eigenvector along a given coordinate. Heuristically, an "important" coordinate will give a larger contribution to the "important" eigenvectors (i.e., with larger eigenvalues).

This quantity might be used to create a hierarchy among the coordinates and, when choosing the manifold to cover, it can offer a useful criteria to decide which quantities to include in the bank. For example, in the manifold `logMq_s1xyz_s2z_iotaphi`, the variable $\log_{10} M$ has an importance of 5×10^4, while variables q, s_1, θ_1 and s_{2z} have importance two orders of magnitude less. This implies that a template bank must include (besides the total mass) all the variables q, s_1, θ_1, and s_{2z}. On the other hand, coordinates ϕ_1 (controlling the magnitude of s_{1y}) and the angles ι and φ have an "importance" of one order of

[11] We neglect any spin on the lighter object, a neutron star. This is physically motivated by the fact that a NS is expected to have low or no spins.

magnitude less than all the other quantities. As a consequence, the latter three play a smaller role in covering the space and they can be possibly ignored (or perhaps only one of them can be included).

Of course, this line of reasoning is heuristic and whether a manifold is suitable or not to cover the space must be checked by means of an injection study. However, the study of the relative importance between coordinates can give an educated guess on the manifold to cover and serve as a starting point for the trial and error process of manifold selection.

c. A proposal for the stochastic template placement Our normalizing flow finds an obvious application within a stochastic placement algorithm. According to the stochastic algorithm, template proposals are randomly drawn from an analytical PDF, which is specifically design to approximate Eq. (15) in the non-spinning case. A good proposal is crucial to reduce the template rejection rate, hence reducing the overall run time.

The normalizing flow is a natural candidate for a proposal distribution, since it goes beyond the nonspinning BBH approximation, allowing for more physics to be considered. Implementing a normalizing flow within the stochastic algorithm will most likely provide a computational benefit, due to a more efficient proposal.

d. Generating datasets for machine learning applications The recent years have seen a burst of machine learning application to GW data analysis, covering all fields of the analysis of compact binary systems from waveform modeling [72–75] to GW searches [101–104] and parameter estimation [105–109].

For all these applications, it is crucial to have high quality datasets of waveforms for training purposes. The goodness and the applicability of the model strongly relies on the distribution of waveforms in the dataset and substantial time is often spent in tuning the dataset composition to achieve optimal performance. The waveforms in such datasets can be sampled using our normalizing flow model, thus covering the space accurately. In many cases this may prove beneficial.

VI. FUTURE PROSPECTS

Clearly, our work can be improved and expanded in several directions. In this section, we discuss some possible advancements.

a. Introducing a new metric As shown in Appendix B, the Hessian of the match (with which we identify the metric) does not always approximate the behavior of the true match in a neighborhood of a point. For instance, on the manifold `Mq_s1xyz`, consider the ellipse $\mathcal{E}_0$, centered on $\theta_0 = (10M_\odot, 7, 0.6, 2, 2)$ of all the points θ with *metric* match with the center higher than 0.97. It turns out that only $\sim$50% of the points inside $\mathcal{E}_0$ have a match higher than 0.97. The situation gets worse for smaller mass ratio, when the metric determinant vanishes, and it can significantly vary among different manifolds.

While this hasn't affected (too much) the effectiveness of our template bank, the failure of the metric approximation is concerning and can negatively influence the placement, especially in presence of a parameter with a small impact on the waveform. The interested reader is encouraged to read Appendix B.

b. Exploring different flow architectures In this work, we only considered masked autoregressive layers for our normalizing flow architecture. Of course, other choices are available in the literature and could possibly improve the flow accuracy. Further work should implement some of these and assess the (possible) gain in accuracy. Possible transformations include coupling layers [110,111] or residual flows [112,113].

As discussed in Sec. II B, it is very beneficial to use a transformation like Eq. (19) as the first layer of the normalizing flow. Future work can find a different transformation offering better performance.

c. Estimating the covering fraction with importance sampling An accurate evaluation of the covering fraction in Eq. (25) is crucial to providing a realistic estimation of the template number and hence good coverage. Currently we estimate the covering fraction by using the approximation to the volume element given by the normalizing flow. We can increase the accuracy by computing the integral in Eq. (25) with importance sampling:

$$\hat{\eta}(\mathcal{T}) \simeq \frac{1}{\sum_i w_i} \sum_i c(\theta_i) w_i \tag{35}$$

where the livepoints are sampled from the flow and are weighted with weights $w_i = \frac{\sqrt{|M(\theta_i)|}}{p^{\mathrm{flow}}(\theta_i)}$. The weights make sure that we evaluate the unapproximated version of the integral, i.e., using the true volume element and not its flow approximation.

In a practical application, it is wise to prevent the weights to grow indefinitely, as this can negatively impact the estimation of the covering fraction. For this reason, we clip the weights to a maximum value of W: $w_i = \min\left(\frac{\sqrt{|M(\theta_i)|}}{p^{\mathrm{flow}}(\theta_i)}, W_{\max}\right)$. The tuning of $W_{\max}$ deserves more attention, as it can really impact the bank performance.

Some tests have shown that importance sampling delivers larger banks, thus with better coverage but with an increased variance in the number of templates. However, in some occasions, one or a few livepoints can dominate the sum (i.e., have very large weight), making the covering fraction computation less robust in case of flow inaccuracies. More work is required to treat such cases and successfully implement this new feature.

d. Exploring different placement methods While the random template placement method in use has proven its efficacy, other alternatives are certainly possible. A different placement method is appealing to reduce the bank size without degrading its performance, as random template banks tend to place more templates than needed.

First, one could use the metric to reject templates that are too close to each other. This would be a variation of the stochastic algorithm, where distances are computed with the metric and not with the true match. While this may prove unfeasibly slow in some cases, it can still be computationally more efficient than with the brute force match computation. As a compromise, a random template bank with low covering fraction and minimal match might be given as starting point for the iteration (i.e., a seed bank).

One could also devise alternative strategies to sample from the flow latent space, such as using quasi Montecarlo sampling or even setting points on a lattice. Since the coordinates of the templates will be correlated with each other, we cannot compute iteratively the covering fraction as described in Sec. II C. For this reason the suitable bank size needs to be computed with other methods, before selecting the templates.

Regardless of the placement method, the templates in a bank may still not be placed optimally, creating over (under)-dense regions. This is especially true for the *random* method used here. For this reason, it may be beneficial to add a postprocessing step to move or remove some templates [114].

e. Encoding the metric into the flow? A fascinating path to explore is to encode information about the metric M_{ij} inside the flow transformation. So far, the normalizing flow ϕ_W is trained in such a way that the determinant of the Jacobian $\det J_{\phi_W}$ matches the determinant of the metric. Thus, among the $\frac{D(D-1)}{2}$ free components of J_{ϕ_W}, only one of them is constrained during the training. This leaves a lot of degeneracy in J_{ϕ_W}. One could break such degeneracy by imposing the additional constraint that the Jacobian of the flow matches the metric M_{ij}:

$$(J_{\phi_W})_{ij} \simeq M_{ij}. \tag{36}$$

Such constraint should be imposed by introducing a suitable loss function. The approach would involve a much harder optimization problem and it remains to be assessed whether the flow has enough representation power to solve such problem.

A flow trained in this way would create an isometry (i.e., distance preserving transformation) between the latent space and the physical space. According to differential geometry, this is not possible, unless the M_{ij} has zero curvature, which is not the case in general. A possible way out could be to embed the manifold of signals in an higher dimensional flat manifold, which would guarantee the existence of a solution.

As outlined, there are many open questions and issues to solve, which require significant work. The reward however would be huge: the flow would parameterize a distance preserving (and not only volume preserving) transformation, which can be used for high dimensional fast stochastic placement or even geometric placement—the holy grail of bank generation.

VII. FINAL REMARKS

We present a novel method to generate template banks covering a high-dimensional manifold of (possibly) precessing/HM/eccentric BBH signals.

Key to our method is the metric M_{ij} and the derived volume element $\sqrt{|M|}$. The latter defines the number of templates that should cover an infinitesimal volume and can be seen as a probability measure on the space. We derive here for the first time an expression for the metric suitable for precessing and/or HM signals (see Appendix A). The metric is written in terms of the gradients of the waveform polarizations and is numerically stable.

To sample the templates, we introduce a novel normalizing flow model, which serves the twofold purpose of sampling from the space and providing a fast-to-compute approximation to $\sqrt{|M|}$. Once we are able to sample from the space, we place templates using the *random* algorithm, which is fast and suitable to cover high-dimensional spaces. This comes at the price of a larger bank than would be produced with the state-of-the-art stochastic algorithm, although the over-coverage becomes less severe as the number of dimensions, and correspondingly the overall size of the bank, increases.

We validate our code by evaluating the normalizing flow accuracy and the robustness of the random placement. Moreover, with a few hours of computation, we were able to reproduce two template banks existing in the literature obtained with independent codes—a nonspinning HM bank [21] and an aligned-spin bank [56].

To demonstrate the capabilities of our code, we generate two large template banks covering systems for which no or little searches have been performed: a precessing bank gathering 1.6 million templates (Sec. V A) and an aligned-spin HM bank formed by 2.1 million templates (Sec. V B). We show that the two banks satisfactorily cover the space. They were both produced in a matter of hours, with minimal CPU and memory usage. We also discuss other possible applications of our method, including the optimization of the template proposal of the stochastic algorithm, the selection of a suitable parameter space for a GW search and the generation of datasets of waveforms for the training of machine learning models.

Our code is publicly available as a package MBANK [55] and comes with a large number of tools to simplify the bank generation and validation.

As a final remark, we stress that our work will enable the GW community to run searches on novel regions of the

BBH parameter space. Being able to generate a high dimensional bank in a few hours, the computational cost of searching new regions of the parameter space will be dominated by the actual cost of the analysis rather than the cost of prior steps. This will allow for optimal resource allocation to search for signatures of precession, eccentricity and/or HMs, hopefully leading to exciting physics discoveries.

ACKNOWLEDGMENTS

We thank Melissa Lopez Portilla, Harsh Narola, Aaron Zimmerman, and Keith Riles for their precious comments. We should not forget to thank the anonymous referee who stimulated huge improvements to our work with their interesting comments. S. S., B. G., and S. C. are supported by the research program of the Netherlands Organization for Scientific Research (NWO). S. C. is also supported by the National Science Foundation under Grant No. PHY-2309332. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by the National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation.

APPENDIX A: DETAILS OF THE METRIC COMPUTATION

In this appendix we report the details of the derivation of Eq. (12), as well as the computation of the Hessian H of the overlap in Eq. (7) in terms of the gradients of the waveform $h(\theta)$. In what follows, we define $(h_1|h_2)$ and $[h_1|h_2]$ to be the real and imaginary part, respectively, of $\langle h_1|h_2\rangle$.

We begin by expanding the quantity $\mathcal{M}(\theta+\Delta\theta,\theta)$ for $\Delta\theta$ around 0. Since $\mathcal{M}(\theta+\Delta\theta,\theta)$ has a maximum for $\Delta\theta=0$, the leading term is quadratic in $\Delta\theta$. We obtain:

$$\begin{aligned}\mathcal{M}(\theta+\Delta\theta,\theta) &= \max_{\Delta t}\mathcal{O}(\theta+\Delta\theta,\theta,\Delta t)\\ &= \max_{\Delta t}\Big\{1+\frac{1}{2}[\partial_{ij}\mathcal{O}\Delta\theta_i\Delta\theta_j\\ &\quad + 2\partial_{it}\mathcal{O}\Delta\theta_i\Delta t+\partial_{tt}\mathcal{O}(\Delta t)^2]\Big\}\\ &= 1+\frac{1}{2}\left[\partial_{ij}\mathcal{O}-\frac{\partial_{it}\mathcal{O}\partial_{jt}\mathcal{O}}{\partial_{tt}\mathcal{O}}\right]\Delta\theta_i\Delta\theta_j \qquad \text{(A1)}\end{aligned}$$

where all the derivatives are evaluated at $\Delta\theta=\Delta t=0$ and the explicit time maximization yields $\Delta t=-\frac{\partial_{it}\mathcal{O}\Delta\theta_i}{\partial_{tt}\mathcal{O}}$.

From Eq. (A1), we can read the expression for the metric in Eq. (12) recognizing in the derivatives $\partial\partial\mathcal{O}|_{\Delta\theta,\Delta t=0}$ the components of the Hessian matrix H of the overlap.

We now compute the Hessian H of the overlap in terms of the gradients of the *normalized* waveforms. For notational convenience, we set $h_+(\theta_1)e^{ift}=s$, we drop any dependence on θ_2 and we understand $\mu=i,t$. We have:

$$\begin{aligned}\partial_\mu\mathcal{O} = \frac{1}{\mathcal{O}}\frac{1}{1-\hat{h}^2_{+\times}}&[(\partial_\mu\hat{s}|\hat{h}_+)(\hat{s}|\hat{h}_+)+(\partial_\mu\hat{s}|\hat{h}_\times)(\hat{s}|\hat{h}_\times)\\ &-(\partial_\mu\hat{s}|\hat{h}_+)(\hat{s}|\hat{h}_\times)h_{+\times}-(\partial_\mu\hat{s}|\hat{h}_\times)(\hat{s}|\hat{h}_+)h_{+\times}] \qquad \text{(A2)}\end{aligned}$$

Differentiating another time, after some rearrangements, we get:

$$H_{tt}=-(\hat{h}_+|\hat{h}_+f^2)+\frac{1}{1-\hat{h}^2_{+\times}}[\hat{h}_\times|\hat{h}_+f]^2 \qquad \text{(A3)}$$

$$H_{ti}=[\hat{h}_+|\partial_i\hat{h}_+f]-\frac{1}{1-\hat{h}^2_{+\times}}(\hat{h}_\times|\partial_i\hat{h}_+)[\hat{h}_\times|\hat{h}_+f] \qquad \text{(A4)}$$

$$H_{ij}=(\hat{h}_+|\partial_i\partial_j\hat{h}_+)+\frac{1}{1-\hat{h}^2_{+\times}}(\hat{h}_\times|\partial_i\hat{h}_+)(\hat{h}_\times|\partial_j\hat{h}_+) \qquad \text{(A5)}$$

To move further, we express the normalized waveform derivatives in terms of the unnormalized ones:

$$\text{(i)}\ \partial_i\langle h|h\rangle=\langle\partial_i h|h\rangle+\langle h|\partial_i h\rangle=2(h|\partial_i h)$$

$$\text{(ii)}\ \partial_i\hat{h}=\frac{1}{(h|h)^{3/2}}[(h|h)\partial_i h-(h|\partial_i h)h]$$

$$\text{(iii)}\ \partial_t\hat{h}=if\hat{h}=if\frac{h}{(h|h)^{1/2}}$$

$$\text{(iv)}\ \partial_i\partial_j\hat{h}=\frac{1}{(h|h)^{1/2}}\partial_{ij}h+3\frac{1}{(h|h)^{5/2}}(h|\partial_i h)(h|\partial_j h)h-\frac{1}{(h|h)^{3/2}}[(h|\partial_{ij}h)h+(\partial_i h|\partial_j h)h+2(h|\partial_{(i}h)\partial_{j)}h]$$

where $A_{(ij)}=\frac{1}{2}(A_{ij}+A_{ji})$ denotes symmetrization.

Plugging this into the Eqs. (A3)–(A5), we get:

$$H_{tt} = -\frac{1}{h_{++}}(h_+|f^2h_+) + \frac{1}{1-\hat{h}^2_{+\times}}\frac{1}{h_{++}h_{\times\times}}[h_\times|fh_+]^2 \tag{A6}$$

$$H_{ti} = -\frac{1}{h_{++}}(h_+|f\partial_i h_+) - \frac{1}{1-\hat{h}^2_{+\times}}\frac{1}{h_{++}h_{\times\times}}[h_\times|fh_+](h_\times|\partial_i h_+) + \frac{\hat{h}_{+\times}}{1-\hat{h}^2_{+\times}}\frac{1}{h^{3/2}_{++}h^{1/2}_{\times\times}}[h_\times|fh_+](h_+|\partial_i h_+) \tag{A7}$$

$$\begin{aligned} H_{ij} = &-\frac{1}{h_{++}}(\partial_i h_+|\partial_j h_+) + \frac{1}{1-\hat{h}^2_{+\times}}\frac{1}{h^2_{++}}(h_+|\partial_i h_+)(h_+|\partial_j h_+) + \frac{1}{1-\hat{h}^2_{+\times}}\frac{1}{h_{++}h_{\times\times}}(h_\times|\partial_i h_+)(h_\times|\partial_j h_+) \\ &-\frac{2\hat{h}_{+\times}}{1-\hat{h}^2_{+\times}}\frac{1}{h^{3/2}_{++}h^{1/2}_{\times\times}}(h_\times|\partial_{(i}h_+)(h_+|\partial_{j)}h_+) \end{aligned} \tag{A8}$$

where we defined $h_{\cdot *} = (h_\cdot|h_*)$.

Such expressions, together with Eq. (12) fully specify the metric. The gradients $\partial_i h$ of the waveform can be computed with a finite difference scheme or analytically for a number of surrogate waveform models [72–75].

The nonprecessing limit can be recovered by setting $h_\times = ih_+$ and $h_{+\times} = 0$:

$$H_{tt} = \frac{1}{h^2_{++}}(h_+|fh_+)^2 - \frac{1}{h_{++}}(h_+|f^2h_+) \tag{A9}$$

$$H_{ti} = \frac{1}{h^2_{++}}[h_+|\partial_i h_+](h_+|h_+f) - \frac{1}{h_{++}}[h_+|f\partial_i h_+] \tag{A10}$$

$$\begin{aligned} H_{ij} = &\frac{1}{h^2_{++}}\{(h_+|\partial_i h_+)(h_+|\partial_j h_+) + [h_+|\partial_i h_+][h_+|\partial_j h_+]\} \\ &-\frac{1}{h_{++}}(\partial_i h_+|\partial_j h_+) \end{aligned} \tag{A11}$$

APPENDIX B: ALTERNATIVE DEFINITIONS FOR THE METRIC

Throughout this paper, we identified the metric with the Hessian of the overlap [see Eq. (12)]. While this is widely used in the literature [44,45] and has been proven to provide reliable template banks, it still has some undesirable properties. To show this, we compute the metric at point $\theta_0 = (20M_\odot, 3, 0.7, 1.8)$ of manifold `Mq_s1xz`, described in Sec. III A, and we compute its eigenvalues $\alpha^{(i)}$ and eigenvectors $v^{(i)}$. We then compute the match $\mathcal{M}^{(i)}_\epsilon$ between θ_0 and the point $\theta^{(i)}_\epsilon = \theta_0 + \epsilon v^{(i)}$, located at a distance ϵ along ith eigenvector. Finally, we compute the coefficient α of the Taylor expansion $1 - \mathcal{M}^{(i)}_\epsilon = \alpha\epsilon^2$. α corresponds to the ith eigenvalue and in principle, it should be close to its value.

In Fig. 15, we plot the fitted relation between $1 - \mathcal{M}$ and ϵ for each eigenvector, as well as the one computed with the metric. In the legend we report the α coefficient (dashed blue line) and the eigenvalue of the metric (solid orange line). The striking feature we note in Fig. 15 is that the eigenvalue is consistently smaller than the fitted α coefficient, sometimes by an order of magnitude. This means that the Hessian, which is computed for $\epsilon \to 0$, is not able to extrapolate the behavior of $1 - \mathcal{M}(\epsilon)$ even at modestly large value of ϵ: the metric approximation to the match loses its predictivity as a measure of distance. The problem becomes more severe in high-dimensional manifolds. On the other hand, since the banks generated with the Hessian metric show nice coverage, one may argue that the *volume* estimate provided by the Hessian is still accurate enough for our purposes.

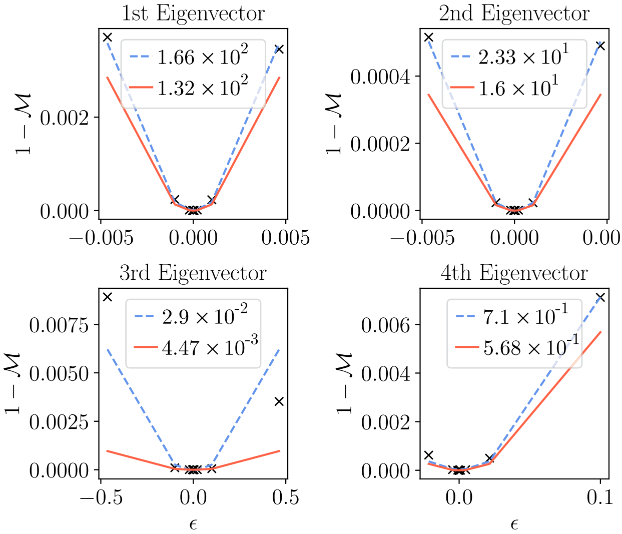

FIG. 15. For each eigenvector of the metric, we compute the empirical relation between the mismatch $1 - \mathcal{M}$ and the distance ϵ of points along the eigenvector direction. The solid line shows the relation predicted by the metric, while the dashed line shows a parabolic fit. In the legend are reported the quadratic coefficients of both lines.

As a way out, we could redefine the matrix $M_{ij}(\theta)$ to a more suitable expression, departing from the Hessian. The goodness of the metric expression may depend on the application and on the range of validity of the approximation. The tensor field $M_{ij}(\theta)$ can be computed through an optimization problem, where we minimize the discrepancy between the two quantities in Eq. (11), encoded into a *loss function*. The loss function depends on the values of the matrix elements M'_{ij}:

$$\mathcal{L}_\theta(M'_{ij}) = \int_{\{d(\theta,\theta')<d_{\text{target}}\}} d^D\theta' [1 - \mathcal{M}(\theta,\theta') - M'_{ij}\Delta\theta_i\Delta\theta_j]^2 \tag{B1}$$

where the integration extends on a D-ball with radius d_{target} centered around θ and d_{target} is a tunable parameter, which controls the validity of the approximation.

At any given point θ, the components $M_{ij}(\theta)$ of the metric are selected by minimizing the above loss:

$$M_{ij}(\theta) = \underset{M'_{ij}}{\operatorname{argmin}}\, \mathcal{L}_\theta(M'_{ij}). \tag{B2}$$

Although the minimization can be tackled with standard techniques, it requires many evaluations of Eq. (10) and the ability to sample from a "complex" set such as $\{d(\theta,\theta') < d_{\text{target}}\}$.

While in most cases this may prove unfeasible, future work could solve the problem in Eq. (B2) at a manageable cost. This may be beneficial to many data analysis applications, such as template placement and Fisher information matrix studies. A number of alternative metric expressions, coming from different heuristic optimization strategies, are already available in MBANK, although not fully validated.

APPENDIX C: COMPUTING THE VOLUME OF THE PARAMETER SPACE

As the number of templates is proportional to the volume of the parameter space [44], it can be useful to estimate the volume of the parameter space. This can be useful to forecast the size of a template bank. The volume can be easily estimated by *importance sampling* and, as the normalizing flow reproduces the volume element, it is a convenient distribution to generate samples.

The volume of the parameter space $\mathcal{B}_D$ is defined as:

$$\mathcal{V} = \int_{\mathcal{B}_D} d^D\theta \sqrt{\det M(\theta)} \tag{C1}$$

$$= \int_{\mathcal{S}_{\text{flow}}} d^D\theta \sqrt{\det M(\theta)}\, \mathcal{I}_{\mathcal{B}_D}(\theta) \tag{C2}$$

where in the last equality we compute the integral on the support of the flow $\mathcal{S}_{\text{flow}} \supseteq \mathcal{B}_D$ and we introduced the indicator function $\mathcal{I}_{\mathcal{B}_D}$ which is non-zero only on the manifold $\mathcal{B}_D$.

Equation (C2) can be numerically evaluated by importance sampling:

$$\mathcal{V} \simeq \frac{1}{N}\sum_i \frac{\sqrt{\det M(\theta_i)}}{p^{\text{flow}}(\theta_i)} \mathcal{I}_{\mathcal{B}_D}(\theta_i) \tag{C3}$$

with $\theta_i \sim p^{\text{flow}}$. The normalizing flow ensures a low variance in the volume estimation.

Equation (C3) involves several metric evaluations, which has some computational cost. To further reduce the computational cost, we can use the fact that, after the training procedure, the flow approximates the volume element as follows:

$$\log p^{\text{flow}} - \log \sqrt{|M|} + C \simeq 0 \tag{C4}$$

where C is the trainable constant appearing in Eq. (21). Hence we can replace $\frac{\sqrt{\det M(\theta_i)}}{p^{\text{flow}}(\theta_i)}$ in Eq. (C3) simply with e^C. The volume estimation is then reduced to computing the fraction of the volume of $\mathcal{S}_{\text{flow}}$ covered by $\mathcal{B}_D$:

$$\mathcal{V} \simeq e^C \frac{1}{N}\sum_i \mathcal{I}_{\mathcal{B}_D}(\theta_i) \tag{C5}$$

where again $\theta_i \sim p^{\text{flow}}$. The goodness of such approximation is closely related to the flow performance, as studied in Sec. III A (see also Fig. 1).

Once an estimation of the volume is available, the number of templates can be obtained by noting [44] that in a lattice, given a minimal match MM, the average spacing d between template is:

$$d(MM) = 2\sqrt{\frac{1 - MM}{D}} \tag{C6}$$

Hence, roughly speaking, the number of templates N needed to cover the volume $\mathcal{V}$ is given by:

$$N = \frac{\mathcal{V}}{d(MM)^D} \tag{C7}$$

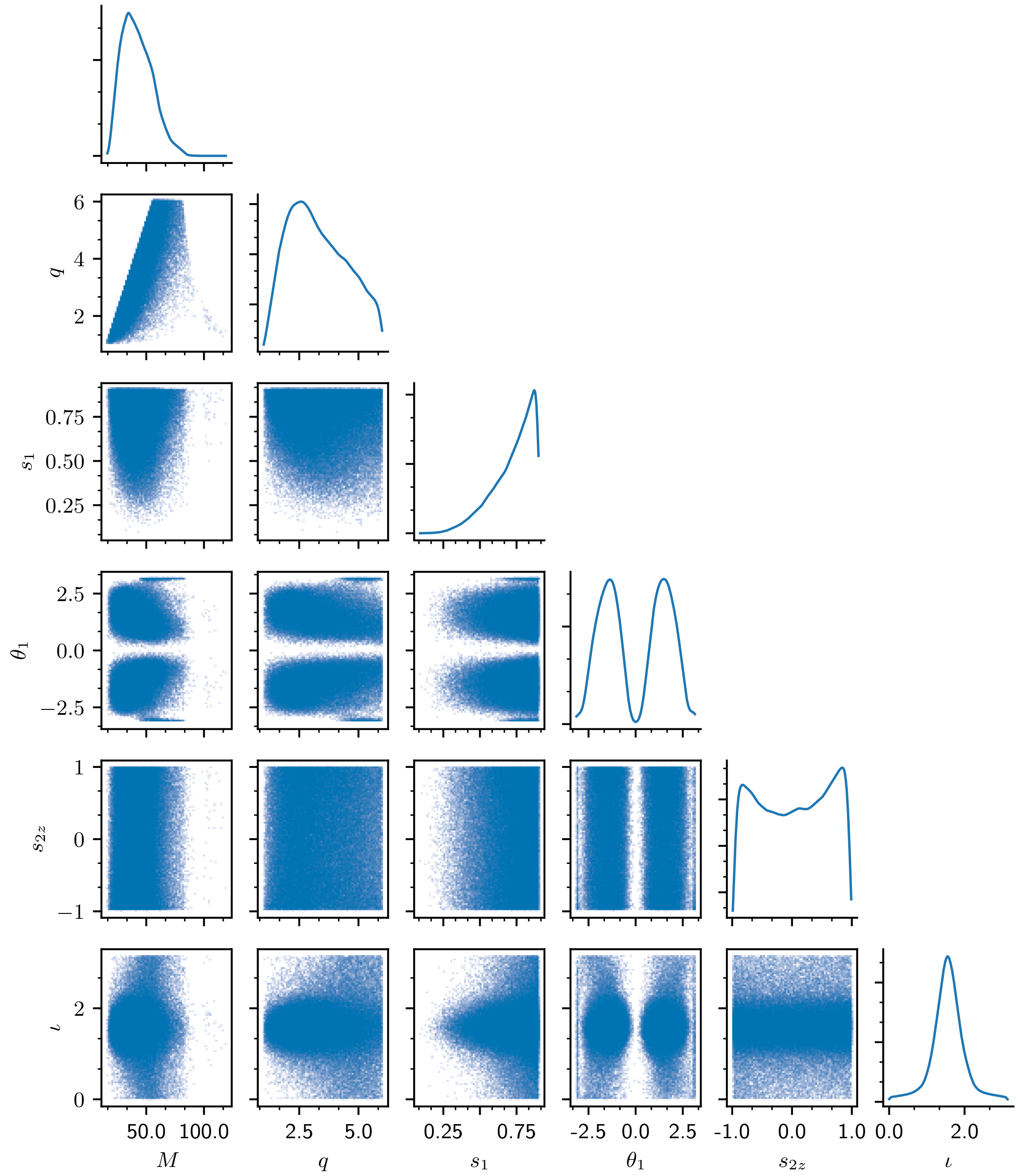

FIG. 16. Corner plot with the templates of the precessing bank described in Sec. V A. Along the diagonals, we show the histogram of the template number as a function of each coordinate.

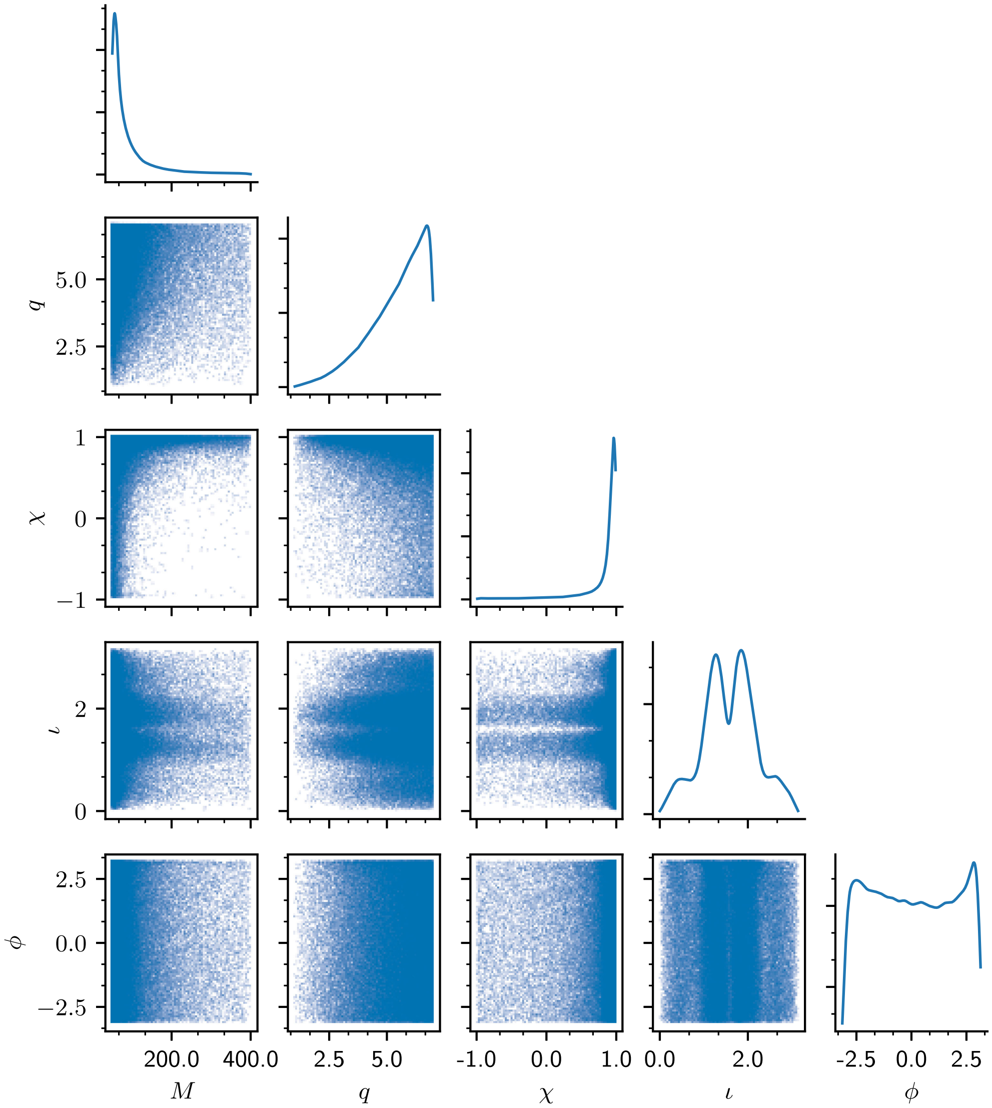

FIG. 17. Corner plot with the templates of the aligned-spin HM bank described in Sec. V B. Along the diagonals, we show the histogram of the template number as a function of each coordinate.

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

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

[3] R. Abbott *et al.*, 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).

[4] R. Abbott *et al.*, 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).

[5] R. Abbott *et al.*, GWTC-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, Phys. Rev. D **109**, 022001 (2024).

[6] R. Abbott *et al.*, GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, Phys. Rev. X **13**, 041039 (2023).

[7] B. P. Abbott *et al.*, Search for subsolar mass ultracompact binaries in Advanced LIGO's second observing run, Phys. Rev. Lett. **123**, 161102 (2019).

[8] R. Abbott *et al.*, Search for subsolar-mass binaries in the first half of Advanced LIGO's and Advanced Virgo's third observing run, Phys. Rev. Lett. **129**, 061104 (2022).

[9] A. H. Nitz and Y.-F. Wang, Broad search for gravitational waves from subsolar-mass binaries through LIGO and Virgo's third observing run, Phys. Rev. D **106**, 023024 (2022).

[10] A. H. Nitz and Y.-F. Wang, Search for gravitational waves from high-mass-ratio compact-binary mergers of stellar mass and subsolar mass black holes, Phys. Rev. Lett. **126**, 021103 (2021).

[11] A. Ramos-Buades, S. Tiwari, M. Haney, and S. Husa, Impact of eccentricity on the gravitational-wave searches for binary black holes: High mass case, Phys. Rev. D **102**, 043005 (2020).

[12] B. D. Cheeseboro and P. T. Baker, Method for detecting highly eccentric binaries with a gravitational wave burst search, Phys. Rev. D **104**, 104016 (2021).

[13] A. H. Nitz, A. Lenon, and D. A. Brown, Search for eccentric binary neutron star mergers in the first and second observing runs of Advanced LIGO, Astrophys. J. **890**, 1 (2019).

[14] B. P. Abbott *et al.*, 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).

[15] A. Ramos-Buades, S. Tiwari, M. Haney, and S. Husa, Impact of eccentricity on the gravitational wave searches for binary black holes: High mass case, Phys. Rev. D **102**, 043005 (2020).

[16] Y.-F. Wang and A. H. Nitz, Prospects for detecting gravitational waves from eccentric subsolar mass compact binaries, Astrophys. J. **912**, 53 (2021).

[17] B. P. Abbott *et al.*, Search for intermediate mass black hole binaries in the first and second observing runs of the Advanced LIGO and Virgo network, Phys. Rev. D **100**, 064064 (2019).

[18] R. Abbott *et al.*, Search for intermediate-mass black hole binaries in the third observing run of Advanced LIGO and Advanced Virgo, Astron. Astrophys. **659**, A84 (2022).

[19] K. Chandra, J. Calderón Bustillo, A. Pai, and I. Harry, First gravitational-wave search for intermediate-mass black hole mergers with higher order harmonics (2022).

[20] I. W. Harry, A. H. Nitz, D. A. Brown, A. P. Lundgren, E. Ochsner, and D. Keppel, Investigating the effect of precession on searches for neutron-star–black-hole binaries with Advanced LIGO, Phys. Rev. D **89**, 024010 (2014).

[21] I. Harry, J. Calderón Bustillo, and A. Nitz, Searching for the full symphony of black hole binary mergers, Phys. Rev. D **97**, 023004 (2018).

[22] S. Fairhurst, R. Green, M. Hannam, and C. Hoy, When will we observe binary black holes precessing?, Phys. Rev. D **102**, 041302 (2020).

[23] N. Indik, K. Haris, T. Dal Canton, H. Fehrmann, B. Krishnan, A. Lundgren, A. B. Nielsen, and A. Pai, Stochastic template bank for gravitational wave searches for precessing neutron-star–black-hole coalescence events, Phys. Rev. D **95**, 064056 (2017).

[24] I. Harry, S. Privitera, A. Bohé, and A. Buonanno, Searching for gravitational waves from compact binaries with precessing spins, Phys. Rev. D **94**, 024012 (2016).

[25] S. Fairhurst, R. Green, C. Hoy, M. Hannam, and A. Muir, Two-harmonic approximation for gravitational waveforms from precessing binaries, Phys. Rev. D **102**, 024055 (2020).

[26] C. McIsaac, C. Hoy, and I. Harry, A search technique to observe precessing compact binary mergers in the advanced detector era, Phys. Rev. D **108**, 123016 (2023).

[27] J. Calderón Bustillo, S. Husa, A. M. Sintes, and M. Pürrer, Impact of gravitational radiation higher order modes on single aligned-spin gravitational wave searches for binary black holes, Phys. Rev. D **93**, 084019 (2016).

[28] K. Chandra, J. C. Bustillo, A. Pai, and I. W. Harry, First gravitational-wave search for intermediate-mass black hole mergers with higher-order harmonics, Phys. Rev. D **106**, 123003 (2022).

[29] C. Mills and S. Fairhurst, Measuring gravitational-wave higher-order multipoles, Phys. Rev. D **103**, 024042 (2021).

[30] D. Wadekar, T. Venumadhav, A. K. Mehta, J. Roulet, S. Olsen, J. Mushkin, B. Zackay, and M. Zaldarriaga, A new approach to template banks of gravitational waves with higher harmonics: Reducing matched-filtering cost by over an order of magnitude, arXiv:2310.15233.

[31] 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).

[32] S. V. Dhurandhar and B. S. Sathyaprakash, Choice of filters for the detection of gravitational waves from coalescing binaries. 2. Detection in colored noise, Phys. Rev. D **49**, 1707 (1994).

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

[34] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries, Phys. Rev. D **85**, 122006 (2012).

[35] 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).

[36] T. Cokelaer, A template bank to search for gravitational waves from inspiralling compact binaries. II. Phenomenological model, Classical Quantum Gravity **24**, 6227 (2007).

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

[38] I. W. Harry, B. Allen, and B. S. Sathyaprakash, A stochastic template placement algorithm for gravitational wave data analysis, Phys. Rev. D **80**, 104014 (2009).

[39] I. W. Harry, B. Allen, and B. S. Sathyaprakash, Stochastic template placement algorithm for gravitational wave data analysis, Phys. Rev. D **80**, 104014 (2009).

[40] P. Ajith, N. Fotopoulos, S. Privitera, A. Neunzert, 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).

[41] T. Dal Canton and I. W. Harry, Designing a template bank to observe compact binary coalescences in Advanced LIGO's second observing run, arXiv:1705.01845.

[42] D. Mukherjee *et al.*, Template bank for spinning compact binary mergers in the second observation run of Advanced LIGO and the first observation run of Advanced Virgo, Phys. Rev. D **103**, 084047 (2021).

[43] A. K. Lenon, D. A. Brown, and A. H. Nitz, Eccentric binary neutron star search prospects for cosmic explorer, Phys. Rev. D **104**, 063011 (2021).

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

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

[46] D. A. Brown, I. Harry, A. Lundgren, and A. H. Nitz, Detecting binary neutron star systems with spin in advanced gravitational-wave detectors, Phys. Rev. D **86**, 084017 (2012).

[47] D. Keppel, Metrics for multi-detector template placement in searches for short-duration nonprecessing inspiral gravitational-wave signals, Phys. Rev. D **86**, 123010 (2012).

[48] S. Roy, A. S. Sengupta, and P. Ajith, Effectual template banks for upcoming compact binary searches in Advanced-LIGO and Virgo data, Phys. Rev. D **99**, 024048 (2019).

[49] J. Roulet, L. Dai, T. Venumadhav, B. Zackay, and M. Zaldarriaga, Template bank for compact binary coalescence searches in gravitational wave data: A general geometric placement algorithm, Phys. Rev. D **99**, 123022 (2019).

[50] A. Coogan, T. D. P. Edwards, H. S. Chia, R. N. George, K. Freese, C. Messick, C. N. Setzer, C. Weniger, and A. Zimmerman, Efficient gravitational wave template bank generation with differentiable waveforms, Phys. Rev. D **106**, 122001 (2022).

[51] C. Hanna *et al.*, Binary tree approach to template placement for searches for gravitational waves from compact binary mergers, Phys. Rev. D **108**, 042003 (2023).

[52] T. Cokelaer, Gravitational waves from inspiralling compact binaries: Hexagonal template placement and its efficiency in detecting physical signals, Phys. Rev. D **76**, 102004 (2007).

[53] B. Allen, Performance of random template banks, Phys. Rev. D **105**, 102003 (2022).

[54] B. Allen, Optimal template banks, Phys. Rev. D **104**, 042005 (2021).

[55] S. Schmidt, MBANK—Metric bank generation for gravitational waves data analysis, https://mbank.readthedocs.io/en/latest/.

[56] S. Sakon *et al.*, Template bank for compact binary mergers in the fourth observing run of Advanced LIGO, Advanced Virgo, and KAGRA, arXiv:2211.16674.

[57] J. Creighton and W. Anderson, *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis* (Wiley, Hoboken, New Jersey, 2011).

[58] M. Maggiore, *Gravitational Waves. Vol. 1: Theory and Experiments*, Oxford Master Series in Physics (Oxford University Press, New York, 2007).

[59] B. S. Sathyaprakash and B. F. Schutz, Physics, astrophysics and cosmology with gravitational waves, Living Rev. Relativity **12**, 2 (2009).

[60] P. Ajith, S. Babak, Y. Chen, M. Hewitson, B. Krishnan, A. M. Sintes, J. T. Whelan, B. Brügmann, P. Diener, N. Dorband, J. Gonzalez, M. Hannam, S. Husa, D. Pollney, L. Rezzolla, L. Santamaría, U. Sperhake, and J. Thornburg, Template bank for gravitational waveforms from coalescing binary black holes: Nonspinning binaries, Phys. Rev. D **77**, 104017 (2008).

[61] S. Privitera, S. R. P. Mohapatra, P. Ajith, K. Cannon, N. Fotopoulos, M. A. Frei, C. Hanna, A. J. Weinstein, and J. T. Whelan, Improving the sensitivity of a search for coalescing binary black holes with nonprecessing spins in gravitational wave data, Phys. Rev. D **89**, 024003 (2014).

[62] S. A. Usman *et al.*, The PyCBC search for gravitational waves from compact binary coalescence, Classical Quantum Gravity **33**, 215004 (2016).

[63] C. Capano, I. Harry, S. Privitera, and A. Buonanno, Implementing a search for gravitational waves from binary black holes with nonprecessing spin, Phys. Rev. D **93**, 124007 (2016).

[64] C. Messick *et al.*, Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data, Phys. Rev. D **95**, 042001 (2017).

[65] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Detecting binary compact-object mergers with gravitational waves: Understanding and improving the sensitivity of the PyCBC search, Astrophys. J. **849**, 118 (2017).

[66] S. 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.

[67] 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).

[68] Q. Chu *et al.*, SPIIR online coherent pipeline to search for gravitational waves from compact binary coalescences, Phys. Rev. D **105**, 024023 (2022).

[69] C. Capano, Y. Pan, and A. Buonanno, Impact of higher harmonics in searching for gravitational waves from nonspinning binary black holes, Phys. Rev. D **89**, 102003 (2014).

[70] P. Schmidt, F. Ohme, and M. Hannam, Towards models of gravitational waveforms from generic binaries II: Modelling precession effects with a single effective precession parameter, Phys. Rev. D **91**, 024043 (2015).

[71] A. J. K. Chua, C. R. Galley, and M. Vallisneri, Reduced-order modeling with artificial neurons for gravitational-wave inference, Phys. Rev. Lett. **122**, 211101 (2019).

[72] S. Khan and R. Green, Gravitational-wave surrogate models powered by artificial neural networks, Phys. Rev. D **103**, 064015 (2021).

[73] S. Schmidt, M. Breschi, R. Gamba, G. Pagano, P. Rettegno, G. Riemenschneider, S. Bernuzzi, A. Nagar, and W. Del Pozzo, Machine learning gravitational waves from binary black hole mergers, Phys. Rev. D **103**, 043020 (2021).

[74] L. M. Thomas, G. Pratten, and P. Schmidt, Accelerating multimodal gravitational waveforms from precessing compact binaries with artificial neural networks, Phys. Rev. D **106**, 104029 (2022).

[75] J. Tissino, G. Carullo, M. Breschi, R. Gamba, S. Schmidt, and S. Bernuzzi, Combining effective-one-body accuracy and reduced-order-quadrature speed for binary neutron star merger parameter estimation with machine learning, Phys. Rev. D **107**, 084037 (2023).

[76] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, J. Mach. Learn. Res. **22**, 64 (2021).

[77] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., Red Hook, 2019), Vol. 32.

[78] I. Kobyzev, S. J. Prince, and M. A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2021).

[79] G. Papamakarios, Neural density estimation and likelihood-free inference, arXiv:1910.13233.

[80] M. Germain, K. Gregor, I. Murray, and H. Larochelle, MADE: masked autoencoder for distribution estimation, in *Proceedings of the 32nd International Conference on Machine Learning* (2015), Vol. 37, pp. 881–889, arXiv:1502.03509.

[81] G. Papamakarios, T. Pavlakou, and I. Murray, Masked autoregressive flow for density estimation, arXiv:1705.07057.

[82] C. Huang, D. Krueger, A. Lacoste, and A. C. Courville, Neural autoregressive flows, arXiv:1804.00779.

[83] S. Khan, S. Husa, M. Hannam, F. Ohme, M. Pürrer, X. Jiménez Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era, Phys. Rev. D **93**, 044007 (2016).

[84] G. Pratten *et al.*, Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes, Phys. Rev. D **103**, 104056 (2021).

[85] LIGO Scientific Collaboration, LIGO algorithm library—LALSuite, free software (GPL) (2018).

[86] C. García-Quirós, M. Colleoni, S. Husa, H. Estellés, G. Pratten, A. Ramos-Buades, M. Mateu-Lucena, and R. Jaume, Multimode frequency-domain model for the gravitational wave signal from nonprecessing black-hole binaries, Phys. Rev. D **102**, 064002 (2020).

[87] R. Abbott *et al.*, Advanced LIGO anticipated sensitivity curves, https://dcc.ligo.org/LIGO-T0900288/public.

[88] X. Zhu, E. Thrane, S. Oslowski, Y. Levin, and P. D. Lasky, Inferring the population properties of binary neutron stars with gravitational-wave measurements of spin, Phys. Rev. D **98**, 043002 (2018).

[89] R. Abbott *et al.*, Noise curves used for simulations in the update of the observing scenarios paper, https://dcc.ligo.org/LIGO-T2000012/public.

[90] L. M. Thomas, P. Schmidt, and G. Pratten, New effective precession spin for modeling multimodal gravitational waveforms in the strong-field regime, Phys. Rev. D **103**, 083022 (2021).

[91] J. Calderón Bustillo, P. Laguna, and D. Shoemaker, Detectability of gravitational waves from binary black holes: Impact of precession and higher modes, Phys. Rev. D **95**, 104038 (2017).

[92] R. Abbott *et al.*, Search for intermediate-mass black hole binaries in the third observing run of Advanced LIGO and Advanced Virgo, Astron. Astrophys. **659**, A84 (2022).

[93] K. Chandra, V. Villa-Ortega, T. Dent, C. McIsaac, A. Pai, I. W. Harry, G. S. C. Davies, and K. Soni, An optimized PyCBC search for gravitational waves from intermediate-mass black hole mergers, Phys. Rev. D **104**, 042004 (2021).

[94] L. Blackburn *et al.*, The LSC Glitch Group: Monitoring noise transients during the fifth LIGO science run, Classical Quantum Gravity **25**, 184004 (2008).

[95] M. Zevin *et al.*, Gravity spy: Integrating Advanced LIGO detector characterization, machine learning, and citizen science, Classical Quantum Gravity **34**, 064003 (2017).

[96] B. P. Abbott *et al.*, Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914, Classical Quantum Gravity **33**, 134001 (2016).

[97] D. Davis *et al.*, LIGO detector characterization in the second and third observing runs, Classical Quantum Gravity **38**, 135014 (2021).

[98] S. Babak, H. Grote, M. Hewitson, H. Luck, and K. A. Strain, Signal based vetoes for the detection of gravitational waves from inspiralling compact binaries, Phys. Rev. D **72**, 022002 (2005).

[99] L. Pekowsky, J. Healy, D. Shoemaker, and P. Laguna, Impact of higher-order modes on the detection of binary black hole coalescences, Phys. Rev. D **87**, 084008 (2013).

[100] V. Varma, P. Ajith, S. Husa, J. C. Bustillo, M. Hannam, and M. Pürrer, Gravitational-wave observations of binary black holes: Effect of nonquadrupole modes, Phys. Rev. D **90**, 124004 (2014).

[101] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Convolutional neural networks: A magic bullet for gravitational-wave detection?, Phys. Rev. D **100**, 063015 (2019).

[102] M. B. Schäfer, F. Ohme, and A. H. Nitz, Detection of gravitational-wave signals from binary neutron star mergers using machine learning, Phys. Rev. D **102**, 063015 (2020).

[103] M. B. Schäfer, O. Zelenka, A. H. Nitz, F. Ohme, and B. Brügmann, Training strategies for deep learning gravitational-wave searches, Phys. Rev. D **105**, 043002 (2022).

[104] G. Baltus, J. Janquart, M. Lopez, A. Reza, S. Caudill, and J.-R. Cudell, Convolutional neural networks for the detection of the early inspiral of a gravitational-wave signal, Phys. Rev. D **103**, 102003 (2021).

[105] S. R. Green, C. Simpson, and J. Gair, Gravitational-wave parameter estimation with autoregressive neural network flows, Phys. Rev. D **102**, 104057 (2020).

[106] J. D. Álvares, J. A. Font, F. F. Freitas, O. G. Freitas, A. P. Morais, S. Nunes, A. Onofre, and A. Torres-Forné, Gravitational-wave parameter inference using deep learning (2020), 10.1109/CBMI50038.2021.9461893.

[107] M. J. Williams, J. Veitch, and C. Messenger, Nested sampling with normalizing flows for gravitational-wave inference, Phys. Rev. D **103**, 103006 (2021).

[108] J. Langendorff, A. Kolmus, J. Janquart, and C. Van Den Broeck, Normalizing flows as an avenue to studying overlapping gravitational wave signals, Phys. Rev. Lett. **130**, 171402 (2023).

[109] M. J. Williams, J. Veitch, and C. Messenger, Importance nested sampling with normalising flows, Mach. Learn. Sci. Tech. **4**, 035011 (2023).

[110] L. Dinh, D. Krueger, and Y. Bengio, Nice: Non-linear independent components estimation, arXiv:1410.8516.

[111] L. Dinh, J. N. Sohl-Dickstein, and S. Bengio, Density estimation using real NVP, arXiv:1605.08803.

[112] Y. Bengio and S. Bengio, Modeling high-dimensional discrete data with multi-layer neural networks, in *Advances in Neural Information Processing Systems*, edited by S. Solla, T. Leen, and K. Müller (MIT Press, Cambridge, MA, 1999), Vol. 12.

[113] J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen, Invertible residual networks, in *Proceedings of the International Conference on Machine Learning* (PMLR, 2018).

[114] N. Indik, H. Fehrmann, F. Harke, B. Krishnan, and A. B. Nielsen, Reducing the number of templates for aligned-spin compact binary coalescence gravitational wave searches using metric-agnostic template nudging, Phys. Rev. D **97**, 124008 (2018).