

## Quantum algorithm for gravitational-wave matched filtering

Sijia Gao<sup>1</sup>,\* Fergus Hayes<sup>1</sup>,† Sarah Croke<sup>1</sup>, Chris Messenger<sup>1</sup>, and John Veitch<sup>1</sup>  
 SUPA, School of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, United Kingdom

 (Received 3 September 2021; accepted 7 March 2022; published 4 April 2022)

Quantum computational devices currently under development have the potential to accelerate data analysis techniques beyond the ability of any classical algorithm. We propose the application of a quantum algorithm for the detection of unknown signals in noisy data. We apply Grover's algorithm to matched filtering, a signal processing technique that compares data to a number of candidate signal templates. In comparison to the classical method, this provides a speedup proportional to the square root of the number of templates, which would make possible otherwise intractable searches. We demonstrate both a proof-of-principle quantum circuit implementation and a simulation of the algorithm's application to the detection of the gravitational wave signal GW150914. We discuss the time complexity and space requirements of our algorithm as well as its implications for the currently computationally limited searches for continuous gravitational waves.

DOI: [10.1103/PhysRevResearch.4.023006](https://doi.org/10.1103/PhysRevResearch.4.023006)

### I. INTRODUCTION

Quantum computing holds enormous potential for computational speedup of certain tasks, offering the possibility of solving classically intractable problems, in particular, in quantum chemistry and many-body physics [1,2]. The technology has seen rapid development in the last few years, resulting in processors with 50–100 qubits, and the first demonstrations of clear quantum advantage over classical computation [3,4]. Quantum algorithms (see Ref. [5] for an accessible overview) are being explored for more and more fields of endeavor: for example, finance [6], quantum simulation [7], particle physics [8,9], machine learning [10,11], and, as the technology matures and a new generation of software developers adopt quantum programming languages, it may be anticipated that new and unexpected applications will be discovered. A particularly versatile quantum subroutine is Grover's search algorithm [12], which finds a marked solution in a large unstructured database. Grover's algorithm, one of the earliest proposed quantum algorithms, provides a square-root speedup over classical search. This is less dramatic than the exponential speedup promised by, e.g., Shor's algorithm [13], but can nevertheless provide a significant practical advantage for problems with a large search space. By defining the search space and conditions for a desired solution, Grover's algorithm may be applied to any computational problem with a limited structure and has found use in minimum finding [14], clustering and nearest-neighbor algorithms for supervised and

unsupervised learning [15,16], and pattern matching [17–19] to name but a few. In this paper, we propose the use of Grover's search in quantum algorithms for matched filtering, with applications in gravitational wave (GW) astronomy. These algorithms inherit the square root speedup of Grover's search algorithm, an improvement which could enable GW searches currently intractable with state-of-the-art classical techniques.

Matched filtering is a signal processing technique [20] in which an exhaustive search is performed over a bank of templates to find the template that when correlated with the data returns the highest detection statistic [21], making it a natural candidate for a quantum speedup through Grover's algorithm. In GW matched-filtering a geometric definition of distance within the parameter space is defined based on the relative loss in signal-to-noise ratio (SNR) between a template and a potential signal. The required distribution of the templates in the search space are chosen so the distance (or overlap) between adjacent templates is constant throughout the space. Depending on the specific data analysis problem, the number of templates can range up to  $\sim 10^{12}$  [22], resulting in a total computational time of  $\sim 10^6$  CPU hours. The spacing of templates in the parameter space determines the efficiency of the search but also the overall number of templates, and the sensitivity of searches for certain classes of signals (e.g., continuous wave sources) is currently computationally limited. Thus, even a modest square-root speedup could enable the detection of signals which would be infeasible with classical techniques.

Key to our proposed algorithms is the fact that the potential signals in GW astronomy are well-modeled by general relativity, and the templates may be readily computed as part of the matching procedure. This eliminates the need to preload the database into quantum random access memory (qRAM) [23], and thus avoids hidden complexity associated with this loading step, as well as doubts about the experimental feasibility of constructing qRAM [24–27]. The presented

\*s.gao.2@research.gla.ac.uk

†f.hayes.1@research.gla.ac.uk

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

algorithms may be applied to any matched filtering problem in which the required templates may be efficiently computed, although we focus here on the application to GW detection. A range of quantum algorithms for data processing and more general learning tasks exist in the literature (e.g., Refs. [10,11,15,16,28–34]). Most closely related to our work are existing algorithms for pattern matching [17–19], which search for an exact or approximate match for a specified pattern (bit string) within a larger data set; these, however, require the data and pattern to be loaded into memory, which would have prohibitive space requirements in the case considered here. Alternatively, algorithms for quantum template matching were first proposed almost 20 years ago [35,36], in which optimal strategies for determining the closest matching template are given. These rely on generalized quantum measurements with one outcome for each possible template; translating a GW template bank into such a measurement is not trivial for the simplest cases, and likely infeasible for the more interesting cases. A related task in the literature is estimating the overlap between quantum states, provided a number of copies of each [37,38]. In GW data analysis, however, the number of templates is by far the largest parameter, and such an approach does not obviously offer an advantage.

Although current state-of-the-art quantum processors are still too small and error prone for many applications of interest, there is much effort concentrated around developing applications for so-called noisy intermediate-scale quantum (NISQ) devices [25], with quantum machine learning being one promising area [10,11,26]. The next technological hurdle will be to implement error correction, and this comes with an overhead in the number of physical qubits required to produce a smaller number of error-corrected logical qubits [39,40]. In the longer run, fully scalable, fault-tolerant devices will be required for universal quantum computation, and to run algorithms such as Shor’s famous factoring algorithm [13]. At this point, further applications in machine learning, pattern matching, and data processing may be expected, to which we now add matched filtering for GW data analysis.

In the remainder of the paper, we show how to employ Grover’s algorithm and its extension to quantum counting to perform quantum matched filtering. We choose a digital encoding for the data and templates, that is, each is encoded as classical bits in the computational basis, and explicitly construct a quantum oracle which returns whether a template matches with the data above a given threshold. We present two algorithms demonstrating the application of quantum counting to matched filtering; the first determines whether there is at least one matching template and provides an estimate to their number; the second returns matching templates. We require only that there is an efficient classical algorithm to generate the templates from an index into the considered set of parameters, and to perform template matching. We discuss the complexity of our algorithms compared to classical techniques, and the implications for GW data analysis. We go beyond an asymptotic analysis to compare the approximate number of matching calculations needed in the classical and quantum algorithms for particular match-filtering problems and defined performance requirements, showing orders of magnitude of difference between the quantum and classical algorithms.

Throughout, it is our aim to present our ideas in a form accessible to both the GW and quantum computing communities. Thus, we provide some background and details to each which will be well-known to experts within each field but may be unfamiliar to the other subset of the intended audience. In Sec. II, we review GWs, matched filtering, Grover’s algorithm, and quantum counting. Following this, we present our algorithm in Sec. III. We give an implementation on IBM’s Qiskit platform [41] in Sec. IV and an analysis of the application to the detection of the first GW detected, GW150914, in Sec. V. We detail the potential speedup provided by our algorithm for matched filtering applied to continuous waves in Sec. VI and discuss the implications to their discovery. We conclude with a discussion of the implications of our paper and suggest directions for further study. We also include an introduction to quantum computing concepts in Appendix A and some of the mathematical details in Appendix B.

## II. BACKGROUND

### A. Gravitational wave searches

The detection of GWs from the merger of compact binary systems is now a regular occurrence. Since the detection of the binary black hole merger known as GW150914 [42], the Advanced LIGO and Advanced Virgo detectors have detected signals from 50 such systems, including two binary neutron star systems [43]. The individual detections, and the population as a whole, allow us to infer properties of GW sources including the nature of extreme matter constituting neutron stars [44], set stringent constraints on the accuracy of general relativity [45], resolve the mystery of the origin of short gamma-ray bursts [46], probe the formation history of compact objects [47], and make measurements on cosmological parameters independent of the cosmic distance ladder [48].

While searches are ongoing for continuously emitted GWs, supernovae and unmodeled burst sources, and the astrophysical and cosmological stochastic backgrounds, as yet only signals from compact binary coalescences have been detected. However, as the advanced GW detectors [49–51] increase in sensitivity and additional detectors join the global network [52,53], our reach into the universe grows. With sensitivity to greater cosmic distances, the rate of detections will grow and other intrinsically weaker classes of signal (e.g., continuous GWs) will become detectable (see Ref. [54] for the most recent results from searches for known millisecond pulsars).

Compact binary and continuous GW sources are subject to a matched-filtering search approach [55–59]. This is motivated by the fact that these sources are very well modeled by general relativity. For transient compact binary signals, template waveforms are obtained through post-Newtonian expansion of the orbital dynamics and calibrated against numerical relativity simulations for the merger and ring-down phase [60,61]. The continuous wave case is somewhat simpler since the waveform is expected to be a weak sinusoid generated by rotating neutron stars with nonzero mass quadrupole moments. Such sources will exhibit slowly varying Doppler modulation of the frequency due to the motion of the detector relative to the source, combined with amplitude modulation

produced by the antenna response of the detector as the Earth rotates [62].

An additional continuous wave problem is that of searching for signals from sources that reside in binary systems. This leads to an additional dramatic increase in parameter space volume and the corresponding numbers of templates [63–65]. When comparing the compact binary and continuous wave cases, the relative size of the search spaces, and hence the number of required templates, is typically much greater for the continuous wave case [22]. In fact, the number of templates required for a fully coherent analysis for a continuous wave source of unknown sky location, frequency, and first frequency time derivative (representing the slow drift in the intrinsic spin of the source), makes such a search completely infeasible. Searches such as these are computationally limited in their sensitivity, and so less sensitive but tractable semicoherent approaches are applied. Such schemes subdivide the data in either time or frequency space, analyze each part separately, and then combine the results in such a way as to ignore the signal phase coherence between segments, significantly reducing the computational cost at the expense of sensitivity. To a lesser extent, there are computational limitations for the compact binary searches when extending the search space to preprocessing systems [66] and a coherent analysis between different detectors [67].

### B. Matched filtering

Matched filtering is a signal processing technique used to maximize the SNR by correlating a signal template with measured data. It is the optimal linear method for detecting a known signal buried in Gaussian noise [20] and is close to optimal for the case of searching over a collection of possible templates [21]. For the derivation of a matched filter, consider the detector output time-series to be  $h(t)$ , defined

$$h(t) = s(t) + n(t), \tag{1}$$

where  $s(t)$  is the signal which is added to some noise  $n(t)$ . Now consider a linear filter  $q(t)$  that is applied to the data in the form of an inner product. Assuming the signal has some finite duration, this can be written in the frequency domain denoted  $\tilde{\cdot}$  as

$$\begin{aligned} q \cdot h &= \int_{-\infty}^{\infty} \tilde{q}^*(f) \tilde{h}(f) df \\ &= \int_{-\infty}^{\infty} \tilde{q}^*(f) \tilde{s}(f) df + \int_{-\infty}^{\infty} \tilde{q}^*(f) \tilde{n}(f) df. \end{aligned} \tag{2}$$

It is evident that  $q$  should be chosen as to maximize the inner product with the signal while minimizing the expected inner product with the noise. We can define the optimal SNR after applying the linear filter terms for the case of zero-mean noise using

$$\begin{aligned} \text{SNR}^2 &= \frac{|\int_{-\infty}^{\infty} \tilde{q}^*(f) \tilde{s}(f) df|^2}{E[|\int_{-\infty}^{\infty} \tilde{q}^*(f) \tilde{n}(f) df|^2]} \\ &= 2 \frac{|\int_{-\infty}^{\infty} (S_n^{-1/2}(|f|) \tilde{q}(f))^* (S_n^{-1/2}(|f|) \tilde{s}(f)) df|^2}{\int_{-\infty}^{\infty} S_n(|f|) |\tilde{q}(f)|^2 df}, \end{aligned} \tag{3}$$

where  $E[\dots]$  denotes an expectation value over noise realizations, and  $S_n$  is the single-sided noise power spectral density defined here as

$$\frac{1}{2} S_n(|f|) \delta(f - f') = E[\hat{n}(f) \hat{n}^*(f')], \tag{4}$$

where  $\delta$  is the Dirac delta function. This allows for an upper limit to be placed on the SNR using the Cauchy-Schwarz inequality, constraining it to

$$\text{SNR}^2 \leq 2 \int_{-\infty}^{\infty} S_n^{-1}(|f|) |\tilde{s}(f)|^2 df. \tag{5}$$

This upper bound is achieved for Eq. (3) when the template is proportional to the noise-weighted signal  $\tilde{s}(f)/S_n(f)$ . By further applying the constraint that

$$E\left[\left|\int_{-\infty}^{\infty} \tilde{q}^*(f) \tilde{n}(f) df\right|^2\right] = 1 \tag{6}$$

gives the constant of proportionality and allows us to define the normalized optimal template:

$$\tilde{Q}(f) = \left(\int_0^{\infty} S_n^{-1}(f) |\tilde{s}(f)|^2 df\right)^{-1/2} \tilde{s}(f). \tag{7}$$

Let us define  $\rho(t)$  as the matched filter SNR that is determined by applying Eq. (2) across  $h(t)$  using the optimal template from Eq. (7). The inner product in Eq. (2) can be applied across signal arrival times by instead considering a convolution, resulting in an additional phase component in the definition of the SNR. The matched filter SNR can be maximized over the phase at the time of coalescence  $\phi_0$  by constructing a complex normalized template  $\tilde{Q}_c(f)$  defined as

$$\tilde{Q}_c(f) = \tilde{Q}_{\phi_0=0}(f) + i \tilde{Q}_{\phi_0=\pi/4}(f), \tag{8}$$

where  $\tilde{Q}_{\phi_0=\pi/4}(f)$  is the optimal template but out of phase with  $\tilde{Q}_{\phi_0=0}(f)$  by  $\pi/4$ . The matched filter SNR is calculated from the modulus of Eq. (2):

$$\begin{aligned} \rho(t) &= \left| \int_{-\infty}^{\infty} \frac{\tilde{Q}_c^*(f) \tilde{h}(f)}{S_n(|f|)} e^{2\pi i t f} df \right| \\ &= 2 \left| \int_0^{\infty} \frac{\tilde{Q}_c^*(f) \tilde{h}(f)}{S_n(f)} e^{2\pi i t f} df \right|. \end{aligned} \tag{9}$$

For discretized time-series data of  $M$  time steps separated by  $\Delta t$ ,  $\rho$  as a function of the template and data time offset  $t_j$  becomes

$$\rho(t_j) = \frac{2}{M \Delta t} \left| \sum_{k=1}^{(M-1)/2} \frac{\tilde{Q}_c^*(f_k) \tilde{h}(f_k)}{S_n(f_k)} e^{2\pi i j k / M} \right|. \tag{10}$$

The calculation of  $\rho$  across all  $M$  time steps involves the inverse Fourier transform of the product of the frequency domain signal and template, which has a cost of  $O(M^2)$ . This process can therefore benefit in computational efficiency via the use of the (classical) fast Fourier transform (FFT) algorithm, which has a computational cost of  $O(M \log M)$  [68].

For signal detection, the parameter space of interest is discretized and a list of waveforms is constructed as candidate signal templates. This list of potential waveforms is called the *template bank*. The specific number of required templates and specific locations of each template within the parameter space are the subject of much study in both compact binary

coalescence [55–59,69] and continuous GW fields [58,70,71]. A template is considered a *matched template* if it produces a  $\rho$  greater than some set threshold  $\rho_{\text{thr}}$  at any point in the given time series data. The computational cost of calculating  $\rho$  and comparing the value to  $\rho_{\text{thr}}$  for all  $M$  time steps for a template bank of  $N$  templates is  $O(NM \log M)$ .

### C. Grover’s algorithm

The speedup provided by Grover’s algorithm is proved in an oracle model: The algorithm is given access to an oracle, which returns whether or not a given input is a good match, and in the quantum version it is assumed to allow queries in superposition. Thus, while a classical oracle can test one input at a time, a quantum oracle is capable of acting on a superposition of input values. This means that with a *single* query, a quantum oracle can create a state in the quantum register which is a superposition over many *different* input values, together in each case with the information about whether the input value is a match. Note, however, that this is not immediately useful: a measurement on the register returns just one input value at random, and it is necessary to carefully design procedures to take advantage of this apparent parallelism. For the case of searching a database, this is achieved by Grover’s algorithm. We stress here, however, for readers who do not have a quantum computing background, that each query to the oracle, even when performed on a superposition of many possible input values, really is just one query, with a fixed computational cost independent of the number of input values in the superposition.

One way to achieve this is to assume that the database of interest is preloaded into qRAM [23]. This can be efficiently queried, however there remain doubts about the experimental feasibility of qRAM, as well as whether the advantage over classical techniques persists once all resources needed are taken into account [24–27]. Further, for the problem considered here, the size of the database is prohibitively large, and thus we require an explicit construction of the oracle. There are therefore two requirements for a speedup in a problem of interest: There must be no classical algorithm giving an improvement over a brute force search, and it must be possible to construct an oracle for the problem considered. Further, the oracle should be efficient, meaning that the computational cost of implementing the oracle must scale at most polylogarithmically in the number of entries in the database.

In this section and elsewhere in the paper, we use the asymptotic notation  $O$  and  $\Omega$  common in computing science to discuss the running time or number of gates required. It is assumed that the number of gates scales linearly with the number of operations, so the asymptotic limit of the number of floating point operations of a classical algorithm is comparable to the asymptotic limit of the number of required gates in the quantum case [72]. The statement that  $O(f(N, M))$  gates are required means that the asymptotic scaling of the number of gates required is *upper* bounded by the function  $f(N, M)$  of the parameters  $N, M$  characterizing the size of the input. Similarly,  $\Omega(f(N, M))$  denotes a *lower* bound in the asymptotic scaling. Where possible, we also go beyond asymptotic scaling and give the exact number of operations needed for

particular examples, to illustrate the potential speedup over classical techniques.

Grover’s algorithm, proposed by Lov Grover in 1996 [12], is a quantum algorithm providing a polynomial speedup for search problems compared to classical techniques. A search problem is one in which the aim is to identify one or more *marked entries*, i.e., those satisfying a specified criteria, from within an unstructured database. For a database with  $N$  entries and exactly one marked entry, it is necessary to check  $N/2$  entries on average before finding the marked entry; thus, the required search time for a classical algorithm is  $O(N)$  [73]. Grover’s algorithm finds a solution in  $O(\sqrt{N})$  search time. It was later proved that this is asymptotically optimal;  $\Omega(\sqrt{N})$  queries are required for a quantum algorithm to succeed with high probability [74]. Grover’s algorithm is covered in several introductory quantum computing texts, e.g., Refs. [72,73,75,76], but for the purposes of clarity we use the remainder of this section to outline the algorithm.

We begin with some very brief introductory remarks introducing basic concepts and terminology in quantum computing. The fundamental carrier of quantum information is the qubit, the analogy to the classical bit. Physically, this is a quantum system with two orthogonal states, which we label  $|0\rangle$  and  $|1\rangle$ , and which are known as computational basis states. A quantum register is made up of an array of qubits. Any classical bit string may be encoded into qubits by encoding in the computational basis, simply by preparing  $|0\rangle$  for 0 and  $|1\rangle$  for 1, known as digital encoding. Quantum gates are reversible, due to unitarity of quantum evolution, and any classical reversible logic operation can be directly implemented as a transformation of computational basis states. Note that reversibility is not a restriction, as any classical irreversible computational may be performed reversibly, most straightforwardly by simply retaining copies of the input [73,75,76]. Finally, it is worth stating explicitly that quantum algorithms generically are probabilistic, succeeding with high probability. This is also not a limitation, as the probability of success can be boosted close to one by a few repetitions of the algorithm. Some commonly used states and operations are defined in Appendix A.

Grover’s algorithm establishes a gap in query complexity between classical and quantum computers in an oracle model, that is, it assumes access to an oracle, a black box which computes a desired function but not necessarily a description of the function itself. The query complexity is then given by the number of calls required to the oracle. To cast the search problem as an oracle problem, a function  $f(x)$  is defined which takes the value  $f(x) = 1$  if and only if  $x$  is a marked entry in the database, otherwise  $f(x) = 0$ . In the quantum case, this is implemented by a quantum black box or oracle  $U_f$  that acts as follows on computational basis states:

$$U_f : |x\rangle \otimes |d\rangle \mapsto |x\rangle |d \oplus f(x)\rangle, \quad (11)$$

where  $\otimes$  represents the tensor product and  $\oplus$  is bitwise addition modulo 2. The first register is an input register; the state  $|x\rangle$  represents the input  $x$ , stored as a classical bit string in the computational basis. The second register is an output register; after application of  $U_f$ , the evaluation of the function is contained here, shifted by the initial bitstring  $d$ . The key difference in the quantum case is that the oracle may be

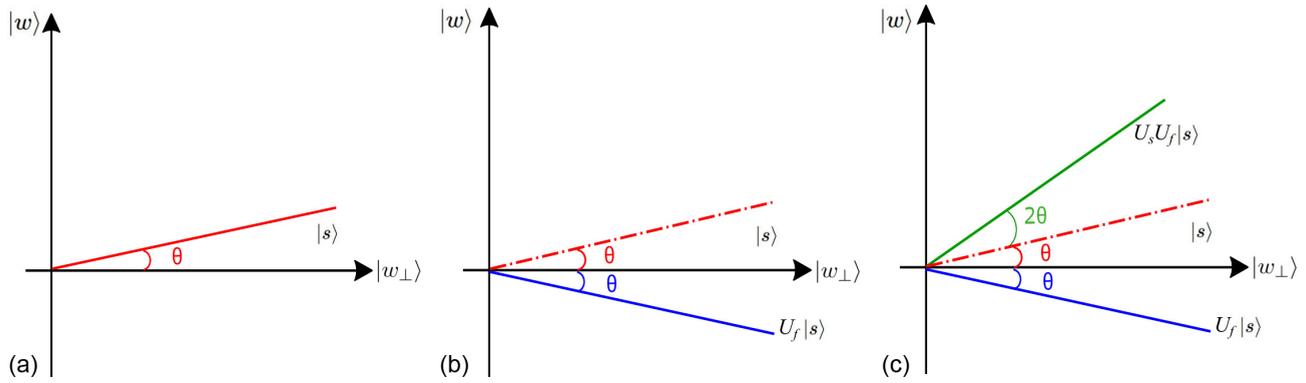


FIG. 1. We show how the input state  $|s\rangle$  changes at different stages of Grover’s algorithm. The two-dimensional space is spanned by the desired match  $|w\rangle$  and undesired match  $|w_{\perp}\rangle$ . The solid lines represent the current state and the dotted lines represent the previous states. (a) The input state, represented by the red line. (b) The state after the oracle is applied, represented by the blue line. (c) The state after the diffusion operator, represented by the green line.

queried in superposition, that is, the input register may be prepared in a superposition over all input states. Note that if the output register is prepared in the state  $|-\rangle$  [see Eq. (A2)], the operation given in Eq. (11) is equivalent to the following procedure, known as phase kickback, on the input register alone:

$$U_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle. \tag{12}$$

Although in the actual algorithm presented later we will need the output register for the oracle, in the following discussion, we prefer to use Eq. (12) for the oracle evaluation for simplicity.

In the problem of searching in an unstructured database, the index of each entry in the database is represented as a computational basis state  $|i\rangle$ , and the input register is prepared in an equal superposition over all indices  $|s\rangle$ . Supposing that there are  $N$  entries, the initial state of the input register can be expressed as

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle, \tag{13}$$

where  $1/\sqrt{N}$  represents the amplitude of each state in the superposition. This corresponds to an equal initial weighting of each entry. State  $|w\rangle$  is used to represent an equal superposition of all the *marked* entries in the database. In the following, we will denote the number of marked entries by  $r$ . The equal superposition of all the other entries of the database is denoted  $|w_{\perp}\rangle$ , which is perpendicular to the state  $|w\rangle$ . In terms of  $|w\rangle$  and  $|w_{\perp}\rangle$ , the input state  $|s\rangle$  may be rewritten as

$$|s\rangle = \sqrt{\frac{r}{N}}|w\rangle + \sqrt{\frac{N-r}{N}}|w_{\perp}\rangle. \tag{14}$$

Now, to increase the probability of finding one of the correct solutions, the next steps of Grover’s algorithm are designed to increase the amplitude of the state  $|w\rangle$  in the superposition. Throughout the algorithm, the state of the input register remains within a real two-dimensional vector space spanned by  $|w\rangle$  and  $|w_{\perp}\rangle$ . The initial state  $|s\rangle$  is shown in Fig. 1(a), where

the angle between the states  $|w\rangle$  and  $|s\rangle$  is defined as

$$\theta = \arcsin(\langle w|s\rangle) = \arcsin\left(\sqrt{\frac{r}{N}}\right). \tag{15}$$

After applying the oracle  $U_f$ , the input state  $|s\rangle$  is transformed to

$$U_f|s\rangle = -\sqrt{\frac{r}{N}}|w\rangle + \sqrt{\frac{N-r}{N}}|w_{\perp}\rangle, \tag{16}$$

which is equivalent to flipping the input state  $|s\rangle$  with respect to the horizontal axis  $|w_{\perp}\rangle$ , as represented in Fig. 1(b). This procedure itself, however, does not make the desired state  $|w\rangle$  more favorable in the measurement. Therefore, an additional diffusion unitary operator is applied as the second step, which is defined as

$$U_s = 2|s\rangle\langle s| - \hat{I}, \tag{17}$$

where  $\hat{I}$  is the identity operator. Considering the state afterward expressed in an orthonormal basis including state  $|s\rangle$ , it is clear that this operator applies a minus sign to the amplitude of all states except  $|s\rangle$ . Analogously to the interpretation of the oracle, this is equivalent to reflecting the state of the register about the equal superposition state  $|s\rangle$ , as shown in Fig. 1(c).

The overall effect of the Grover operator  $\hat{G}$ , defined as

$$\hat{G} = U_s U_f, \tag{18}$$

is shown in Fig. 1(c), and is equivalent to a rotation operator in the two-dimensional space spanned by  $|w\rangle$  and  $|w_{\perp}\rangle$ :

$$\hat{G} = \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix}. \tag{19}$$

After applying the Grover operator  $k$  times, the input state would become

$$\hat{G}^k|s\rangle = \sin((2k+1)\theta)|w\rangle + \cos((2k+1)\theta)|w_{\perp}\rangle, \tag{20}$$

and to maximize the probability of finding one of the desired matches comprising the superposition  $|w\rangle$ , the amplitude  $\sin((2k+1)\theta)$  should be maximized. Thus, the Grover operator is applied  $k$  times such that  $(2k+1)\theta \approx \pi/2$ . This means that if the number  $r$  of matching templates is known, for large

values of  $N/r$ :

$$k \approx \frac{\pi}{4} \sqrt{\frac{N}{r}} - \frac{1}{2}. \tag{21}$$

After  $k$  applications of Grover’s algorithm, as all matching templates are in superposition, a measurement of the input register will return only one of them at random. To obtain additional matching templates the algorithm must be repeated  $r \ln r$  times [77].

**D. Quantum counting**

In many cases, the number of marked entries,  $r$ , is not known in advance. In this case, there exist variants of Grover’s algorithm which return a marked entry with  $O(\sqrt{N/r})$  applications of the oracle [78,79]. The most relevant for our purposes is quantum counting, which uses a well-known primitive in quantum computing, quantum phase estimation [80], to estimate the eigenvalues  $\pm 2\theta$  of the Grover operator introduced in Eq. (19). This, in turn, allows an estimate of  $r$  and of the number of applications of the Grover operator needed to find a solution with high probability.  $O(\sqrt{N})$  Grover iterations are sufficient to determine  $r$  to an accuracy  $O(\sqrt{r})$  with high probability. We complete this background section with an outline of quantum counting, and refer the reader again to Refs. [72,73,75,76] for more information.

Recall that the Grover operator  $\hat{G}$  acts as a rotation in the two-dimensional space spanned by  $|w\rangle$  and  $|w_\perp\rangle$ , as given in Eq. (19). The eigenvectors of  $\hat{G}$  are

$$|s_+\rangle = \begin{pmatrix} \frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \quad |s_-\rangle = \begin{pmatrix} \frac{-i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \tag{22}$$

with eigenvalues of  $e^{2i\theta}$  and  $e^{-2i\theta}$ , respectively, and the input state in Eq. (13) may be written as an equal superposition of the two eigenstates,  $|s_+\rangle$  and  $|s_-\rangle$ :

$$|s\rangle = \frac{1}{\sqrt{2}}(|s_+\rangle + |s_-\rangle). \tag{23}$$

Given an estimate of  $\theta$ , an estimate of the number of matching templates can be obtained through Eq. (15). Therefore, the problem of finding the number of desired templates is transformed into an eigenvalue estimation problem, which can be solved using quantum phase estimation [76]. Phase estimation makes use of the quantum Fourier transform (QFT), which transforms between the computational basis  $\{|j\rangle\}$  and the Fourier basis,  $\{|\tilde{j}\rangle\}$  defined as

$$|\tilde{j}\rangle = \hat{U}_{\text{QFT}}|j\rangle = \sum_{l=0}^{2^p-1} \exp\left(i\frac{2\pi jl}{2^p}\right)|l\rangle. \tag{24}$$

where  $\hat{U}_{\text{QFT}}$  denotes the QFT [75].

In quantum counting, an additional register, which we refer to as the *counting register*, is needed to store the estimate of  $\theta$ . We denote the number of qubits in the register by  $p$ , which we leave unspecified for now. The counting register is first initialized in an equal superposition over all possible

computational basis states:

$$\hat{H}^{\otimes p}|0\rangle^{\otimes p} = \frac{1}{2^{\frac{p}{2}}}( |0\rangle + |1\rangle ) \otimes \dots \otimes ( |0\rangle + |1\rangle ) = \sum_{j=0}^{2^p-1} |j\rangle. \tag{25}$$

Following this, Grover’s operator is applied iteratively to the input state as before, where now the number of applications of the Grover gate is controlled by the counting register:

$$\begin{aligned} & \sum_{j=0}^{2^p-1} C\text{-}\hat{G}^j|j\rangle \otimes |s\rangle \\ &= \frac{1}{\sqrt{2}} \left( \sum_{j=0}^{2^p-1} e^{i2\theta j} |j\rangle \otimes |s_+\rangle + \sum_{j=0}^{2^p-1} e^{-i2\theta j} |j\rangle \otimes |s_-\rangle \right), \end{aligned} \tag{26}$$

where  $C\text{-}\hat{G}^j$  represents applying the controlled Grover’s operator  $j$  times, giving

$$\begin{aligned} & \hat{U}_{\text{QFT}}^{-1} \sum_{j=0}^{2^p-1} C\text{-}\hat{G}^j|j\rangle \otimes |s\rangle \\ &= \frac{1}{2^{p+\frac{1}{2}}} \sum_{j=0}^{2^p-1} \sum_{l=0}^{2^p-1} \left( e^{i2\pi j(\frac{\theta}{\pi} - \frac{l}{2^p})} |l\rangle \otimes |s_+\rangle \right. \\ & \quad \left. + e^{i2\pi j(\frac{\pi-\theta}{\pi} - \frac{l}{2^p})} |l\rangle \otimes |s_-\rangle \right). \end{aligned} \tag{27}$$

A measurement of the counting register in the computational basis returns an integer value between 0 and  $2^p - 1$ , from which we can now extract the desired estimate of the phase. Intuitively, constructive interference occurs for those elements  $\{|l'\rangle\}$  for which

$$\frac{\theta}{\pi} - \frac{l'}{2^p} \simeq 0 \quad \text{or} \quad \frac{\pi - \theta}{\pi} - \frac{l'}{2^p} \simeq 0. \tag{28}$$

We will only be interested in cases in which  $r \ll N$ , and thus  $\theta \ll 1$ . Therefore, the observed measurement outcome, which we denote  $b$ , gives an unambiguous estimate of  $\theta$ , denoted  $\theta_*$  as follows:

$$\theta_* = \begin{cases} \frac{b\pi}{2^p}, & b \leq 2^{p-1} \\ \pi - \frac{b\pi}{2^p}, & b > 2^{p-1}. \end{cases} \tag{29}$$

In reality, values of  $b$  which differ slightly from the constructive interference condition are possible; an example of the probability distribution over  $b$  is shown in Fig. 2. However, it may be shown that the measured value  $b$  gives an estimate of  $\theta$  to  $m$  bits of accuracy with a probability of success at least  $1 - \epsilon$  if  $p$  is chosen such that  $p = m + \log_2(2 + 1/2\epsilon)$  [75]. In quantum counting, an estimate of accuracy at least  $O(N^{-1/2})$  is required, as  $\theta$  itself is of this magnitude. Thus  $m$  and  $p$  are each of size  $1/2 \log_2 N$ . The maximum number of applications of  $\hat{G}$  is given by  $2^p$ , which is therefore  $O(\sqrt{N})$ . From the estimate of  $\theta$ , it is then possible to estimate  $r$  and  $k$ , the number of applications of  $\hat{G}$  needed to subsequently retrieve a marked entry with high probability. In the following sections, we will discuss the choice of  $p$  in more detail for the application to quantum matched filtering, going beyond the asymptotic analysis.

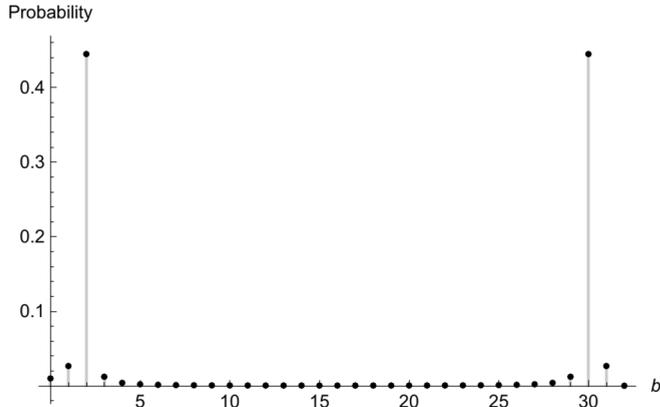


FIG. 2. The probability distribution for each output value in the final measurement on a five-qubit counting register, with two matching entries in a 64-entry database. The two peaks correspond to the two eigenstates defined in Eqs. (22). Constructive interference only happens for values close to  $2^p\theta/\pi$  or  $2^p(\pi - \theta)/\pi$ , with destructive interference occurring elsewhere, resulting in this probability distribution.

### III. QUANTUM MATCHED FILTERING ALGORITHM

In the previous section, we introduced matched filtering, Grover’s algorithm and its extension to quantum counting, and outlined the computational speed-up promised by quantum algorithms for the process of search in an unstructured database. A reminder of the nomenclature that was set in the previous section is provided to the reader in Table I. In this section we argue that matched filtering for GW detection provides a natural application of quantum counting. We detail the pseudo-code of a possible implementation and prove that we can effectively construct the required oracle. We will also compare the computational cost of the quantum approach with the classical cost, taking account of the cost of the oracle evaluation, to evaluate overall complexity in each case and the relative speedup.

As discussed in the previous section, matched filtering involves comparing data (originally) in the form of a time series against templates drawn from a template bank, searching for one or more matches above a predetermined threshold. The templates for GW data analysis are well modeled by general relativity, and rather than performing comparisons against a

TABLE I. Nomenclature used throughout the text to describe variables used to perform Grover’s algorithm and quantum counting, where inferred variables are denoted by the \* subscript.

Variable	Description
$r$	True number of matching templates
$\tilde{b}$	The noninteger counting register value corresponding to $r$
$k$	Number of Grover operations corresponding to $r$
$2\theta$	Rotation in state space corresponding to $r$
$b$	Observed counting register outcome
$r_*$	Number of matching templates corresponding to $b$
$k_*$	Number of Grover operations corresponding to $r_*$
$2\theta_*$	Rotation in state space corresponding to $r_*$

#### Algorithm 1. Grover’s gate complexity: $O(M \log M + \log N)$

```

1: function GROVER’S SEARCH ALGORITHM  $N, |D\rangle, \rho_{\text{thr}}$ 
2:   procedure ORACLE CONSTRUCTION
3:     Creating templates:
4:     for all  $i < N$ , do
5:        $|i\rangle|0\rangle \leftarrow |i\rangle|T_i\rangle$ 
6:     Comparison with the data:
7:      $|i\rangle|D\rangle|T_i\rangle|0\rangle \leftarrow |i\rangle|D\rangle|T_i\rangle|\rho(i)\rangle$ 
8:     if  $\rho(i) < \rho_{\text{thr}}$ , then
9:        $f(i) = 0$ 
10:    else
11:       $f(i) = 1$ 
12:       $|i\rangle|D\rangle|T_i\rangle|\rho(i)\rangle \leftarrow (-1)^{f(i)}|i\rangle|D\rangle|T_i\rangle|\rho(i)\rangle$ 
13:    Disentangling registers:
14:     $(-1)^{f(i)}|i\rangle|D\rangle|T_i\rangle|\rho(i)\rangle \leftarrow (-1)^{f(i)}|i\rangle|D\rangle|T_i\rangle|0\rangle$ 
15:     $(-1)^{f(i)}|i\rangle|D\rangle|T_i\rangle|0\rangle \leftarrow (-1)^{f(i)}|i\rangle|D\rangle|0\rangle|0\rangle$ 
16:  procedure DIFFUSION OPERATOR
17:     $\sum (-1)^{f(i)}|i\rangle \leftarrow \sum (2|i\rangle\langle i| - \hat{I})(-1)^{f(i)}|i\rangle$ 

```

previously populated database, these are calculated as part of the matched filtering procedure. Indeed the number of templates can be so large that precalculating and storing these in a database may have prohibitive memory requirements, even in the classical case. Thus a preloaded database is not necessary for a quantum implementation, avoiding the need for a large amount of data to be loaded into qRAM. Further, the steps needed to construct an oracle which determines whether or not a given template is a match are already part of the classical data analysis, and including these explicitly does not diminish the speedup of the quantum approach, which we outline below.

We note that the cost of an oracle call (i.e., a single SNR calculation) is not negligible; this scales with the observing time period and the frequency bandwidth over which the data is analyzed and must be taken into account in a full complexity analysis. Grover’s algorithm does not speed up this step, and one might wonder whether a more sophisticated approach could give a speedup here also. We return to this in the discussion and compare our quantum counting based approach to related tasks from the literature. What quantum counting can do is improve the dependence of the overall computational cost on the number of templates, making previously intractable searches possible. In particular, as it is the spacing of templates, and therefore the overall number of templates required, that determines the sensitivity of the search, a quantum implementation of matched filtering based on quantum counting promises to enable the detection of signals too weak to detect by classical data processing techniques.

#### A. Oracle construction

We propose two applications of quantum counting to GW matched-filtering: one to determine whether there is a match at all, which is often the problem of interest in GW matched filtering, and the other to retrieve a matching template in the case in which there is at least one match. To apply quantum counting in each case, we first require an oracle to perform matched filtering with a predefined threshold. Thus we begin

by detailing in Algorithm 1 the pseudocode to construct the Grover's gate.

We begin with some preliminaries: Recall that the number of templates is denoted by  $N$ , and the number of data points in the time series by  $M$ . We choose a digital encoding, i.e., to represent the data and templates as classical bits encoded in the computational basis. Standard techniques exist to convert any, in general, irreversible classical logic circuit to a reversible one, which may readily be implemented on a quantum computer by replacing classical reversible gates by their quantum equivalents [72,74]. In general, some scratch space is needed to aid in performing all calculations reversibly. We outline a specific implementation, making use of four registers: one *data register* which must be of size (number of qubits) linear in  $M$ , and one *index register*, which requires  $\log_2 N$  qubits. For intermediate calculations, we specify also one register to hold the computed template, which must be of size linear in  $M$ , and one to hold the computed SNR value, which does not scale with  $N$  or  $M$  and is  $O(1)$ . We discuss the space requirements further in Sec. VII.

The basic element of Grover's algorithm is a search over an index into a database, and an oracle construction must calculate the template from the index  $i$ , proceed to calculate the SNR, and, finally, perform the check against the threshold value. We denote the number of gates needed to compute a template waveform from its parameters by  $k_1$ .<sup>1</sup> As each template consists of  $M$  data points, this takes time linear in  $M$ . The number of gates needed to calculate the SNR between a template and the data is denoted  $k_2$ . From the introduction in Sec. II B, this requires time  $O(M \log M)$ . Finally, checking whether the result is above a given threshold  $\rho_{\text{thr}}$ , as defined in Sec. II B, takes  $O(1)$  gates, and is denoted  $k_3$ . In this way, to compute the match against all templates, we need  $N(k_1 + k_2 + k_3)$  steps, which is the total classical cost. Consequently, the total computational complexity of the classical algorithm is  $O(NM \log M)$ .

To construct a quantum algorithm, we require all the same steps, but in addition we need to erase the intermediate calculations to disentangle the index register from everything else to complete the oracle application. The pseudocode for Grover's gate is given in Algorithm 1.

*Discussion:* The following is the explanation for each step and the related computational cost for Algorithm 1.

*Oracle construction:*

Step 0: Initialization [cost:  $O(M + \log N)$ ]. The initial state is comprised of four registers,

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_I |0\rangle_T |D\rangle_D |0\rangle_\rho, \quad (30)$$

where the subscripts  $I$ ,  $T$ ,  $D$ , and  $\rho$  represent the indices, templates, data, and the SNR register, respectively. Loading the data takes time linear in  $M$ , while initializing the

index register to an equal superposition requires  $O(\log N)$  gates [75].

Step 1 (lines 3–5): Creating templates [cost:  $O(M)$ ]. Calculating the templates from the index is performed in superposition over all index values at a cost of  $k_1 \sim O(M)$  gates. The state after this step would be

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_i^N |i\rangle_I |T_i\rangle_T |D\rangle_D |0\rangle_\rho. \quad (31)$$

Step 2 (lines 6–11): Comparison with the data [cost:  $O(M \log M)$ ]. The cost of calculating SNR between the template and the data is  $k_2 \sim O(M \log M)$ . Finally, we compare this result to a predetermined threshold to determine the value of  $f(i)$ ; the function that determines whether a given template is a match or not at a cost of  $k_3 \sim O(1)$ . After this step, the state becomes

$$|\psi_2\rangle = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{f(i)} |i\rangle_I |T_i\rangle_T |D\rangle_D |0\rangle_\rho. \quad (32)$$

Step 3 (lines 12–14): Disentangling registers [cost:  $O(M \log M)$ ]. The diffusion operator part of Grover's gate must act on the index register alone. If the index register is entangled with any other register, it will not have the desired effect. Therefore, we need to erase the computation of  $\rho(i)$  and  $T_i$  to remove any correlation between these registers and the index register. The erasure process is the reverse of the generation process. Accordingly, another  $k_1 + k_2$  cost is generated. The state after this step is

$$|\psi_3\rangle = \frac{1}{\sqrt{N}} \sum_i^N (-1)^{f_i} |i\rangle_I |0\rangle_T |D\rangle_D |0\rangle_\rho. \quad (33)$$

Step 4 (lines 15 and 16): Applying the diffusion operator [cost:  $O(\log N)$ ]. This step is unique to the quantum algorithm and requires  $O(\log N)$  quantum gates [82].

*Total cost:* The total cost for a single oracle call is therefore

$$O(M \log M + \log N). \quad (34)$$

## B. Signal detection

Now that we have constructed the required oracle for quantum matched filtering, we can readily apply quantum counting to problems of relevance to GW data analysis. Our application will first focus on whether there is a signal existing in the data, a common example in matched filtering. Once it has been identified that a signal is present, a full Bayesian parameter analysis to determine the properties of the source must be performed separately [83,84]. Quantum counting returns  $r_*$ , an estimate of the number of matches, and so is ideally suited to this task.

To identify if there is a signal, we are interested in four conditional probabilities: a *true negative*, the probability of correctly returning that there is no template with an SNR above the predetermined threshold when there is no such template existing in the template bank,  $P(r_* = 0|r = 0)$ ; a *false negative*, the probability of identifying that there is no match when indeed there is no template in the template bank with an SNR above the predetermined threshold,  $P(r_* = 0|r >$

<sup>1</sup>We also need to specify the mapping from index to template parameters. For reasons of clarity, we have not included this step explicitly here, but note that efficient algorithms exist (see Ref. [81]), which add a modest complexity  $O(\text{polylog} N)$ . We discuss template placing in the examples in Secs. V A and VI.

**Algorithm 2.** Signal Detection Complexity:  $O(\sqrt{N}(M \log M + \log N))$

```

1:  $p \leftarrow$  number of precision digits
2:  $N \leftarrow$  number of templates
3:  $i \leftarrow$  index of templates
4:  $\rho_{\text{thr}} \leftarrow$  threshold
5:  $|0\rangle \leftarrow$  Data  $|D\rangle$ 
6: procedure QUANTUM COUNTING ( $p, N, |D\rangle, \rho_{\text{thr}}$ )
7:   Creating the counting register:
8:    $|i\rangle \leftarrow |0\rangle^p |i\rangle$ 
9:    $|0\rangle^p |i\rangle \leftarrow \frac{1}{2^{p/2}} (|0\rangle + |1\rangle)^p \otimes |i\rangle$ 
10:  Controlled Grover's gate:
11:  for all  $j < 2^p$  do
12:     $a \leftarrow j$ 
13:    repeat
14:      Algorithm 1 GROVER'S GATE ( $N, |D\rangle, \rho_{\text{thr}}$ ),  $a - -$ 
15:      until  $a = 0$ 
16:       $\frac{1}{2^{p/2}} (|0\rangle + |1\rangle)^n \otimes |i\rangle \leftarrow$ 
 $\frac{1}{2^{(p+1)/2}} \sum (e^{2i\theta j} |j\rangle \otimes |s_+\rangle + e^{-2i\theta j} |j\rangle \otimes |s_-\rangle)$ 
17:      Inverse quantum Fourier transform:
18:       $\frac{1}{2^{(p+1)/2}} \sum (e^{2i\theta j} |j\rangle \otimes |s_+\rangle + e^{-2i\theta j} |j\rangle \otimes |s_-\rangle) \leftarrow$ 
 $\frac{1}{2^{p+1/2}} \sum \sum (e^{i2\pi j(\frac{a}{\pi} - \frac{1}{2^p})} |l\rangle \otimes |s_+\rangle + e^{i2\pi j(\frac{a-\theta}{\pi} - \frac{1}{2^p})} |l\rangle \otimes |s_-\rangle)$ 
19:      Measurement ( $b$ ):
20:      if  $b = 0$ , then
21:        return 'There is no match.'
22:      else  $r_* \leftarrow \text{Round}[N \sin(\frac{b}{2^p} \pi)^2]$ 
23:      if  $r_* = 0$ , then
24:         $r_* \leftarrow 1$ 

```

0); a *true positive*, the probability of identifying that there are templates with a SNR above the predetermined threshold when there exists such templates in the template bank,  $P(r_* > 0 | r > 0)$ ; and a *false alarm*, the probability of identifying that there are templates with a SNR above the predetermined threshold when there no such template exists it template bank,  $P(r_* > 0 | r = 0)$ . The terms defined here differ from the standard definitions, where the different types of errors depend on whether or not a signal is present in the data and not if templates match. Classical algorithms in signal detection make classification errors due to uncertainty in the data and model, while quantum algorithms have additional uncertainty introduced by their probabilistic nature. It is these classification errors that are made by this additional uncertainty that we are concerned with in this paper. The rate of classification errors due to the data and model uncertainty in this algorithm is the same as the classical case.

Recall that quantum counting returns an integer  $b$ , between 0 and  $2^p - 1$ , from which we can estimate  $\theta$  and therefore  $r$ . If there are no matches, perfect constructive interference occurs for  $b = 0$  in Eq. (27) and  $b = 0$  is returned with certainty. Thus, identifying whether or not there is a signal present simply requires us to check whether  $b = 0$  or  $b \neq 0$ . There will be some probability of returning  $b = 0$  in cases where there are in fact one or more matches, resulting in a false negative output of the algorithm. This may be made exponentially small through a constant number of repetitions. The resulting pseudocode is detailed in Algorithm 2. As discussed earlier,  $2^p$  is required to be  $O(\sqrt{N})$  to give a sufficient accuracy to

distinguish  $\theta$  from zero. At the end of this subsection, we discuss further the impact of the choice of  $p$  on the probability of a false negative.

*Discussion:* The following is the explanation for each step and the related computational cost for Algorithm 2.

*Signal detection:*

Step 0: Initialization [cost:  $O(M + \log N)$ ]. This is the same as step 0 in Algorithm 1.

*Quantum counting:*

Step 1 (lines 7–9): Creating counting register [cost:  $O(\frac{1}{2} \log N)$ ]. This step involves applying a Hadamard gate to each qubit incurring a cost of  $p$ .

Step 2 (lines 10–16): Controlled Grover's gate [cost:  $O((M \log M + \log N)\sqrt{N})$ ]. The cost is given by the largest number of iterations of Grover's gate needed,  $2^p - 1$ .

Step 3 (lines 17 and 18): Inverse quantum Fourier transform (cost:  $O((\log N)^2)$  [73]).

Step 4 (lines 19–24): Measurement [cost:  $O(\frac{1}{2} \log N)$ ]. The cost of measurement is 1 for each counting qubit. For the actual measurement we obtain a value  $b$ . According to Eq. (29), we can calculate an estimate of the number of matching templates  $r_*$  based on Eq. (15). When there is no matching template, the probability of  $b$  being measured as 0 is 1. Therefore, any other observed value of  $b$  resulting in zero matching templates can be disregarded and thus corresponds to an estimate of one matching template.

*Total cost:*

$$O(\sqrt{N}(M \log M + \log N)). \tag{35}$$

We conclude by discussing the effect of the choice of  $p$  on the probability of a false negative, denoted  $\delta_n$ . According to the discussion in Sec. IID,  $p$  can be written as

$$2^p = c\sqrt{N}, \tag{36}$$

and the following discussion is on the choice of the constant  $c$  and its effect on the probability of a false negative. We will use well-known bounds from the literature to motivate a particular choice of  $c$ , and therefore  $p$ . This is not a unique choice but rather a convenient one for which we can readily bound  $\delta_n$ .

To avoid triggering a false negative, the outcome of measurement of the counting register  $b$  should not be 0. According to Ref. [78], if  $\tilde{b}$  is defined as either  $\theta 2^p / \pi$  or  $(2^p - \theta 2^p / \pi)$  (note that this is not in general an integer value), then the measured value  $b$  differs from  $\tilde{b}$  by  $|b - \tilde{b}| \leq 1$  with a probability at least  $8/\pi^2$ . Therefore, choosing  $p$  such that  $\tilde{b} - 1 > 0$  ensures that the probability of a false negative is at most  $1 - 8/\pi^2$ . With this choice, Eqs. (15) and (28) thus give the following restriction on  $p$ :

$$2^p > \pi \sqrt{\frac{N}{r}}. \tag{37}$$

This restriction is most stringent when  $r = 1$ . Therefore, we obtain a lower bound for the choice of number of counting qubits:

$$2^p > \pi \sqrt{N}. \tag{38}$$

With this choice of  $p$  we can obtain a slightly tighter bound on the false negative probability as follows. From Eq. (B3), the probability of a false negative when there exists one or

more templates can be expressed as

$$\delta_n = P(b = 0 | r > 0) = \frac{1}{2^{2p}} \frac{N \sin^2(2^p \theta)}{r} \leq \frac{1}{2^{2p}} \frac{N}{r}. \quad (39)$$

With the choice in Eq. (38), this probability is inversely proportional to  $r$ , and for all  $r$  is bounded by

$$\delta_n < \frac{1}{\pi^2}. \quad (40)$$

We conclude that the signal detection algorithm based on quantum counting has a false alarm probability of 0 under all conditions, and a false negative probability of  $1/\pi^2$ , given the condition in Eq. (38) is met.

If the false negative rate is  $\delta_n$  for each run, by repeating the whole procedure  $\ell$  times, the probability of obtaining  $b = 0$  every time is  $\delta_n^\ell$ . Therefore, the total tolerance of our procedure would be  $\delta_n^\ell < \pi^{-2\ell}$ . With a repetition logarithmic to its tolerance, the total complexity of the procedure is  $O(\ell\pi\sqrt{N})$ .

In GW research, practical applications normally involve between  $10^4$  to  $10^{12}$  templates [22,85]. With the lower bound of the number of templates,  $10^4$ ,  $p$  can be chosen to be 9 according to Eq. (38). In the classical case, the computational cost is approximately  $10^4$  oracle evaluations, while in the quantum case, 512 evaluations suffice for a single run of the signal detection algorithm. There is therefore an order of magnitude difference in cost even for cases with the lowest number of templates. The upper most extreme case that has been analyzed has  $10^{12}$  templates, in which  $p$  would be chosen as 22, resulting in a computational cost of around  $10^7$  oracle evaluations. As a specific example, for a false negative probability of  $\pi^{-12} \simeq 10^{-6}$  (one in a million), a total of  $6 \times 2^{22} \simeq 3 \times 10^7$  evaluations are required. To reduce this to a one in a billion chance of a false negative, nine repetitions of the algorithm are needed, or a total of around  $4.5 \times 10^7$  oracle evaluations. This is orders of magnitude smaller than the classical cost of  $10^{12}$ .

### C. Retrieving matched templates

In the case of a successful signal detection (the identification of 1 or more matching templates), we might wish to further examine its corresponding parameters using (one of) the matching templates. In this section, we will provide a pseudo algorithm to retrieve one or all matching templates.

The procedure to retrieve matching templates is based on Grover's algorithm in Algorithm 1 and the result  $r_*$  of Algorithm 2. This is not the only way to retrieve a matching template given an unknown number of matches [79], but we anticipate that for most applications the signal detection algorithm would run first to determine whether there is any match above threshold. In any potential subsequent attempt to retrieve a matching template, it is then natural to use the estimate  $r_*$  already obtained.

*Discussion:* The following is the explanation for each step and the related computational cost for Algorithm 3. *Template retrieval:*

Step 0 (lines 6 and 7): Calculating the number of repetitions [cost:  $O(1)$ ]. The output  $r_*$  from Algorithm 2 is imported into Algorithm 3, and we then calculate the number of required repetitions of Algorithm 1 from Eq. (21).

**Algorithm 3.** Template retrieval complexity:  $O(\sqrt{N}(M \log M + \log N))$

---

```

1:  $N \leftarrow$  number of templates
2:  $i \leftarrow$  index of templates
3:  $\rho_{\text{thr}} \leftarrow$  threshold
4:  $|0\rangle \leftarrow$  Data  $|D\rangle$ 
5:  $r_* \leftarrow$  number of matched templates
6: Calculating the number of repetitions:
7:  $k_* \leftarrow \text{Round}[\frac{\pi}{4} \sqrt{\frac{N}{r_*}} - \frac{1}{2}]$ 
8: procedure RETRIEVE ONE TEMPLATE
9:   repeat
10:     Algorithm 1 GROVER'S GATE ( $N, |D\rangle, \rho_{\text{thr}}, k_* - -$ 
11:   until  $k_* == 0$ 
12:   Output:
13:      $i_{\text{correct}}$ 

```

---

Procedure 1 (lines 8–13): Retrieve one template [cost:  $O(\sqrt{N}/r_*(M \log M + \log N))$ ]. Grover's algorithm, Algorithm 1, will be repeated  $k_*$  times to achieve the desired template index. The value of  $k_*$  according to our previous discussion will be  $O(\sqrt{N}/r_*)$ .

The total cost of Algorithm 2 and retrieving one template combined is

$$O(\sqrt{N}(M \log M + \log N)\sqrt{N}). \quad (41)$$

Procedure 2 : Retrieve all matched templates. In the case where all the matched templates are required to be found, it is not as trivial as repeating Procedure 1  $r$  (assuming  $r_* \approx r$ ) times because it samples with replacement. It is, instead, a coupon collector problem [77], which requires  $\Theta(r \log r)$  repetitions of Procedure 1. As long as the number of matching templates is small compared with the total number of templates in the bank, the complexity is the same for both procedures.

We conclude this section by discussing the overall probability of failing to return a matched template following this procedure. Note that if this probability is less than 0.5, then with a constant number of repetitions, it can be made negligibly small to ensure successful retrieval of a matched template.<sup>2</sup>

Without loss of generality, we consider in the following analysis only one eigenvalue in Eq. (27), corresponding to  $|s_+\rangle$ . The corresponding probability distribution for different measured values  $b$  is given in Appendix B. In any given run of the procedure, the probability of returning a matched template according to Eq. (20) is therefore given by

$$P(\text{match}) = |\sin((2k_* + 1)\theta)|^2, \quad (42)$$

where  $k_*$  is the number of Grover's applications calculated through Eq. (21) from outcome  $b$  of Algorithm 2 and corresponding estimates  $\theta_*$ ,  $r_*$ . Using Eq. (B2), the overall probability of failing to retrieve a matched template is

<sup>2</sup>There is nothing special about 0.5 here, as long as the probability of failure is bounded away from 1 this is enough; 0.5 is a convenient choice.

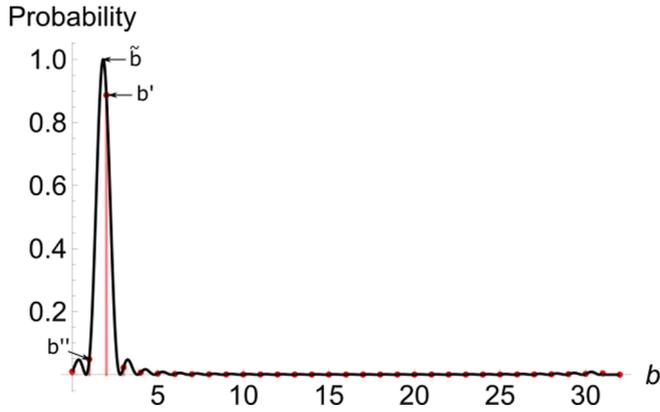


FIG. 3. The red dotted line corresponds to the probability distribution for each state in a five-qubit counting register, with two templates matching in a 64-template bank corresponding to one eigenvalue defined in Eqs. (22). The black line is plotted according to Eq. (B2) as a continuous function. Each peak contains one  $b$  state with a width of 1, except for the central peak which has the two most probable  $b$  states and a width of 2. The upper integer  $b$  state to  $\tilde{b}$  is referred to as  $b'$  with the lower as  $b''$ . The curve peaks at either  $2^p\theta/\pi$  or  $2^p(\pi - \theta)/\pi$  depend on which eigenvalue the curve corresponds to, and is labeled  $\tilde{b}$ .

given by

$$\begin{aligned}
 P(\text{fail}) &= \sum_{l=0}^{2^p} P(\text{fail}|b=l)P(b=l) \\
 &= \frac{1}{2^{2p}} \sum_{l=0}^{2^p} \left( \frac{\sin(2^p\theta)}{\sin(\theta - \frac{\pi l}{2^p})} \right)^2 |\cos((2k_l + 1)\theta)|^2,
 \end{aligned} \tag{43}$$

where  $k_l$  is the number of repetitions of Grover's algorithm when  $b = l$ .

Let  $b'$  be the closest integer larger than  $2^p\theta/\pi$ , i.e.,  $b' = \lceil 2^p\theta/\pi \rceil = 2^p\theta/\pi + \epsilon$ , where  $0 \leq \epsilon \leq 1$ ; and  $b''$  the closest integer smaller than  $2^p\theta/\pi$  such that  $b'' = 2^p\theta/\pi - (1 - \epsilon)$ .  $b'$  and  $b''$  are also the most probable values; recall that the probability that the measured  $b$  value falls into the interval of  $|b - \tilde{b}| \leq 1$  is larger than  $8/\pi^2$  [78]. This is illustrated in Fig. 3 based on Eq. (B2), where the central peak contains the two most probable  $b$  states.

Now an upper bound for  $P(\text{fail})$  is given by only considering the probability of successfully retrieving a template for these two most probable outcomes:

$$\begin{aligned}
 P(\text{fail}) &< P(b')P(\text{fail}|b') + P(b'')P(\text{fail}|b'') \\
 &\quad + (1 - P(b') - P(b'')).
 \end{aligned} \tag{44}$$

Now, to estimate  $P(\text{fail}|b')$ , note using Eq. (21) that

$$\begin{aligned}
 k_{b'} &= \left\lceil \frac{\pi}{4\theta_*} - \frac{1}{2} \right\rceil \\
 &= \frac{\pi}{4\theta_*} - \frac{1}{2} \pm \epsilon_k, \\
 &= \frac{2^{p-2}}{b'} - \frac{1}{2} \pm \epsilon_k,
 \end{aligned} \tag{45}$$

where in the second line  $0 \leq \epsilon_k \leq 0.5$ , and in the third line we have used Eq. (29). In the context of GW searches, i.e.,  $N \gg r$ , the small angle approximation can be applied and consequently,  $\theta \approx \sqrt{r/N}$ . Thus,

$$\begin{aligned}
 (2k_{b'} + 1)\theta &= \frac{2^{p-1}}{b'}\theta \pm 2\epsilon_k\theta \\
 &= \frac{\tilde{b}}{b'} \frac{\pi}{2} + O\left(\sqrt{\frac{r}{N}}\right),
 \end{aligned} \tag{46}$$

from which we obtain using Eq. (42),

$$\begin{aligned}
 P(\text{fail}|b') &= 1 - |\sin((2k_{b'} + 1)\theta)|^2 \\
 &= \left| \cos\left(\frac{\tilde{b}}{b'} \frac{\pi}{2}\right) \right|^2 + O\left(\sqrt{\frac{r}{N}}\right) \\
 &= \left| \cos\left(\frac{b' - \epsilon}{b'} \frac{\pi}{2}\right) \right|^2 + O\left(\sqrt{\frac{r}{N}}\right) \\
 &= \left| \sin\left(\frac{\epsilon}{b'} \frac{\pi}{2}\right) \right|^2 + O\left(\sqrt{\frac{r}{N}}\right).
 \end{aligned} \tag{47}$$

We can also rewrite  $P(b')$  as follows:

$$\begin{aligned}
 P(b') &= \frac{1}{2^{2p}} \left( \frac{\sin(2^p\theta)}{\sin(\theta - \frac{\pi b'}{2^p})} \right)^2 \\
 &= \frac{1}{2^{2p}} \left( \frac{\sin(\tilde{b}\pi)}{\sin(\frac{\pi}{2^p}\epsilon)} \right)^2 \\
 &\simeq \left( \frac{\sin(\epsilon\pi)}{\pi\epsilon} \right)^2,
 \end{aligned} \tag{48}$$

where in the last line we have used the small angle approximation for  $\pi\epsilon/2^p$ , and  $\tilde{b} = b' - \epsilon$ . With similar arguments for  $b''$ , the bound becomes

$$\begin{aligned}
 P(\text{fail}) &< 1 - \left( \frac{\sin(\pi\epsilon)}{\pi\epsilon} \right)^2 \left( \cos\left(\frac{\epsilon}{b'} \frac{\pi}{2}\right) \right)^2 \\
 &\quad - \left( \frac{\sin(\pi(1-\epsilon))}{\pi(1-\epsilon)} \right)^2 \left( \cos\left(\frac{1-\epsilon}{b''} \frac{\pi}{2}\right) \right)^2 \\
 &\quad + O\left(\sqrt{\frac{r}{N}}\right).
 \end{aligned} \tag{49}$$

Recall from Eq. (38), we choose  $p = \lceil \log_2(\pi\sqrt{N}) \rceil$ . It is convenient to express this as  $p = \log_2(\pi\sqrt{N}) + \epsilon_p$ , where  $0 < \epsilon_p < 1$ . Therefore,  $\tilde{b}$  may be written

$$\begin{aligned}
 \tilde{b} &= \frac{2^p\theta}{\pi} \\
 &= \frac{\pi\sqrt{N}2^{\epsilon_p}}{\pi} \sqrt{\frac{r}{N}} \\
 &= 2^{\epsilon_p}\sqrt{r}.
 \end{aligned} \tag{50}$$

Recall that  $b' = \lceil \tilde{b} \rceil$ , and so  $b', \epsilon$  become

$$b' = \lceil 2^{\epsilon_p}\sqrt{r} \rceil; \quad \epsilon = \lceil 2^{\epsilon_p}\sqrt{r} \rceil - 2^{\epsilon_p}\sqrt{r}. \tag{51}$$

Thus for each  $r$  we can write Eq. (49) in terms of a single parameter,  $\epsilon_p$ , between 0 and 1 (neglecting the  $O(\sqrt{r/N})$  term). We optimize this numerically and plot the bound for

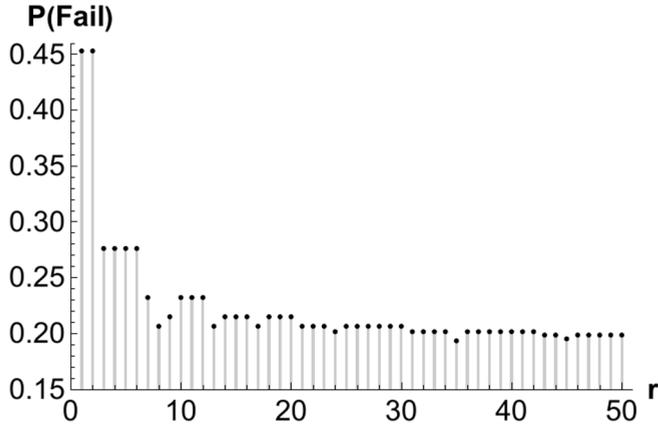


FIG. 4. This shows for large  $N$  the joint probability of obtaining outcome  $b$  and subsequently failing to retrieve a matched template is bounded by 0.45 for different number of matching templates  $r$ .

various values of  $r$  in Fig. 4. In all cases this is less than 0.453, the value found numerically for  $r = 1$ , ensuring the probability of successfully retrieving a template is no smaller than

$$P(\text{success}) \geq 0.547. \quad (52)$$

Note that for large  $r$  (but still requiring  $r \ll N$ ),

$$P(\text{fail}|b') \simeq P(\text{fail}|b'') \simeq \sin^2\left(\frac{1}{\sqrt{r}} \frac{\pi}{2}\right) \simeq O\left(\frac{1}{r}\right),$$

and thus we can expect the bound on the probability of failure to decrease with  $r$  to a limit given by

$$\begin{aligned} P(\text{fail}) &< 1 - P(b') - P(b'') + O\left(\frac{1}{r}\right) \\ &= 1 - \frac{8}{\pi^2} + O\left(\frac{1}{r}\right). \end{aligned} \quad (53)$$

We here provide a specific example of the total probability of failing to retrieve a matching template corresponding to Eq. (43) in Fig. 5. This example has a template bank of  $2^{17}$  templates, with  $r = 9$ , a real GW signal GW150914 that will be discussed in Sec. V. The total failing probability  $P(\text{fail}) \approx 0.34 < 0.5$ . Therefore, with a constant number of repetitions of Algorithms 2 and 3, we are guaranteed with a matched template returned at a complexity of  $O(\sqrt{N}(M \log M + \log N)\sqrt{N})$ . This is less than the classical cost of  $O(NM \log M)$ . Therefore, we conclude that our quantum algorithm offers a  $\sqrt{N}$  speed up with a practical oracle when the number of matching templates is small compared with the total number of templates in the bank.

#### IV. EXAMPLE USING QISKIT

In this section, we will present our proof of principle model of template matching on a quantum computer using IBM's Qiskit library [41] and their quantum computer simulator *ibmq\_qasm\_simulator*.<sup>3</sup> For the uninitiated reader,

<sup>3</sup>The QasmSimulator back end is designed to mimic an actual device. It executes a Qiskit QuantumCircuit and returns a count

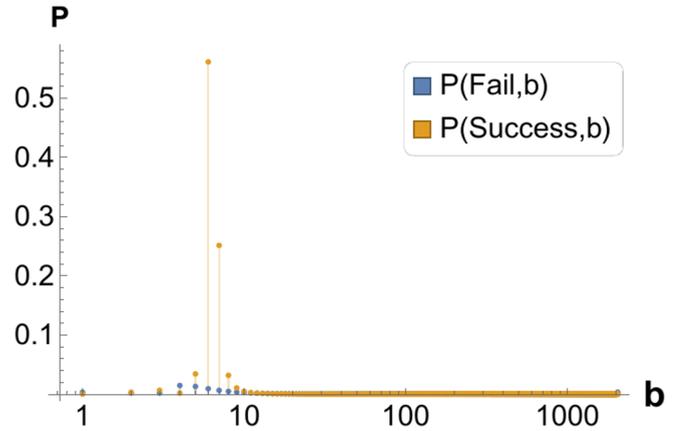


FIG. 5. For the case of a template bank with  $2^{17}$  templates, and  $r = 9$ , the joint probability of obtaining outcome  $b$  and subsequently failing to or succeeding at retrieving a matched template are plotted in blue and yellow, respectively. The total probability of  $P(\text{fail}) \approx 0.34 < 0.5$ .

Appendix A details relevant quantum computing fundamentals that are referred to throughout the following section.

Matching to real GW data requires a much larger quantum processor than is currently available; in Sec. V we will present a classical simulation of matching to actual detector data using Python. Later, we also discuss the space requirements of the matched filtering algorithm. Here, to demonstrate the basic features of a realization on a quantum processor, we implement a simplified algorithm in which we imagine the data is an  $n$ -bit string and the templates are all possible  $n$ -bit strings. This means that the templates themselves are identical to the index, and there is no need to explicitly perform the template generation steps (Algorithm 1, Step 1). We consider that a template is a match to the data if the bit strings are identical, however, to simulate the possibility of nonexact matches, we disregard the  $q$  lowest order bits and require only the  $n - q$  highest order bits to match. The choice of  $q$  is analogous to the choice of threshold SNR value  $\rho_{\text{thr}}$  in the main algorithm. The proof of principle demonstration presented here is thus an example of string matching, a problem considered in Refs. [17–19].

The data consists of an  $n$ -qubit string stored in binary form in the data register  $|D\rangle$ , where the first  $q$  qubits are ignored allowing for  $2^q$  matching templates among  $2^n$  total templates. Hadamard gates are used to initialize the template register  $|T\rangle$  to store a superposition of all possible  $n$ -bit templates. The output qubit  $|d\rangle$  in Eq. (11) is stored in the ancilla register  $|A\rangle$ . An extra counting register with  $p$  qubits is added for the quantum counting procedure.

In our template-matching oracle, which is presented in Fig. 6, we match the template register and the data register qubit-by-qubit using CNOT gates. In the case of an exact match, all the qubits in the template register would be turned into state  $|0\rangle$ . Therefore, after bit flipping, we can use a

dictionary containing the final values of any classical registers in the circuit.

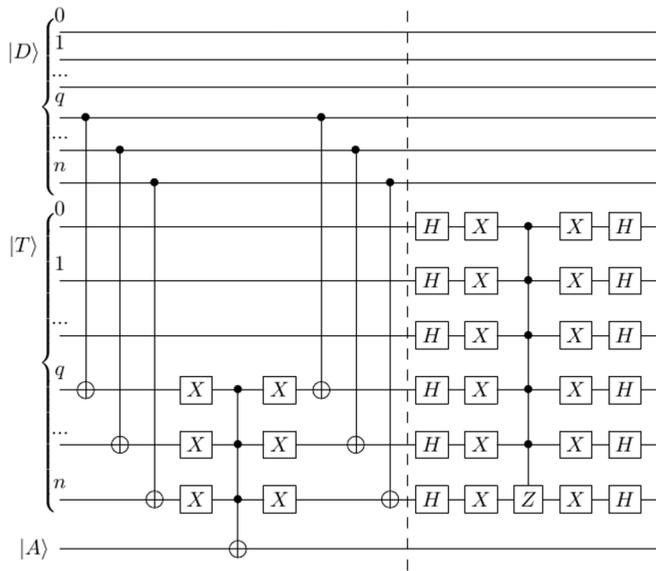


FIG. 6. Quantum circuit diagram for our multiple-template matching oracle and the diffusion operator, which are separated by the vertical dashed line. The  $|D\rangle$  and  $|T\rangle$  variables represent the data and template registers, respectively, and  $|A\rangle$  is the ancilla qubit. The numbers label the  $i$ th qubit in the respective register. To simulate multiple matches, the oracle does not act on the first  $q$  qubits. When there is only one matching template,  $q$  would be 0.

multiple-control-NOT gate to realize phase kickback on the ancillary qubit initialized into the  $|-\rangle$  state. The diffusion operator is constructed by a combination of Hadamard gates, NOT gates, and a  $C^n$ -Z gate, and is illustrated in Fig. 6.

In GW searches, the true signal parameters will lie somewhere within the template bank parameter space and no template will be identical to the signal. Therefore, a predetermined  $\rho_{\text{thr}}$  is chosen as the threshold in Algorithm 1. The number of templates possessing  $\rho$  over this threshold, if there are any, is unknown. Since the optimal number of applications of Grover’s search algorithm is dependent on the number of templates with  $\rho$  over the threshold, we need to apply the quantum counting algorithm first.

To demonstrate a proof of principle of our algorithm, we implement this simplified version with a range of qubits for data and omission, allowing for multiple templates matching. For each pair  $n, q$ , we run the quantum counting algorithm first, in order to estimate the number of matches  $r$ , and then Grover’s algorithm to find a match. From the output of the quantum counting algorithm, we take the most probable value of  $b$  to calculate an estimated  $r_*$  and  $k_*$  for the template retrieval phase. For each algorithm, the experiment is trialed 2048 times and the output of the simulator gives a set of probabilities calculated from the number of occurrences of each possible measured value. The results are presented in Table II. The number of counting qubits,  $p$ , is based on Eq. (38). When the number of qubits for the data,  $n$ , is small,  $p$  is close to  $n$ . However, as  $n$  increases, the difference between  $n$  and  $p$  increases as well, allowing us to maintain the speedup of  $\sqrt{N}$  discussed in Sec. III. The parameters  $k_*$  and  $k$  are the estimated and true number of applications of Grover’s gate needed, given by the quantum counting process by Eqs. (29) and (21) with  $r = 2^q$ , respectively. The probability of the search process returning us with one of the matched templates given the most probable value of  $b$  is over 78% in all cases, and the estimated number of templates,  $r_*$ , differs from the actual number of matched templates,  $2^q$ , by no more than 2.

TABLE II. Trial runs of our algorithm with 2048 iterations on *ibmq\_qasm\_simulator*. We compare the number of iterations Grover’s algorithm should apply and the number of matched templates based on the measured result, to their theoretical counterparts across a range of data with different number of qubits with various number of omitted qubits in the matching process. We also state the  $P(\text{Success})$  as the probability of our algorithm returning us with a matched template in the final search in each case. The number of counting qubits is the minimum allowed by Eq. (38) to minimise the false negative rate,  $\delta_n$ .

ignored qubits $q$	data length $n$	counting qubits $p$	measured count $b$	Grover’s iter. est. $k_*$	est. No. templates $r_*$	Grover’s iter. theo. $k$	$P(\text{Succ.})$
0	5	5	30	4	1	4	0.9995
	6	5	1	6	1	6	0.9961
	7	5	1	8	1	8	0.9956
	8	6	1	12	1	12	1
	9	7	2	17	1	17	0.9990
1	5	5	3	2	3	3	0.9092
	6	5	30	4	2	4	0.9985
	7	6	61	5	3	6	0.9619
	8	6	2	8	2	8	0.9961
	9	7	125	10	3	12	0.9365
2	10	7	126	17	2	17	0.9995
	5	5	4	1	5	2	0.7885
	6	5	29	2	5	3	0.9072
	7	6	60	3	5	4	0.8926
	8	6	61	5	6	6	0.9688
	9	7	124	7	5	8	0.9429
	10	7	125	10	6	12	0.9395

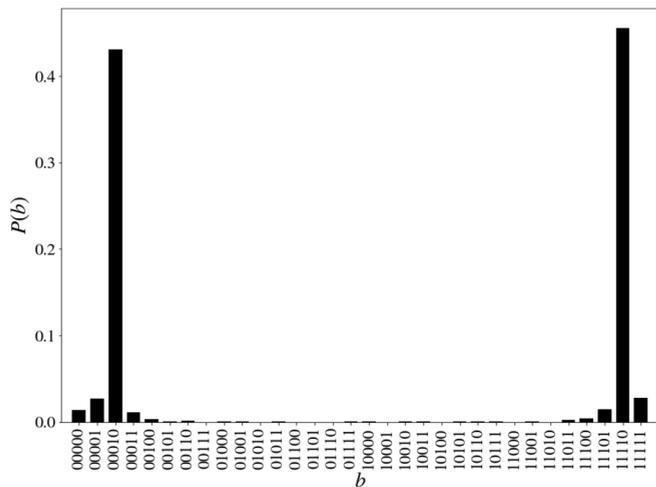


FIG. 7. The measurement of the quantum counting process for six-qubit data matching with a five-qubit counting register. The first qubit is ignored to allow for two templates matching. The theoretically most probable outcome  $b$  in this case, according to Eqs. (28), should be either 2 or 30. The most probable measurement result is 11110, which in decimal is 30.

A specific instance is illustrated in Figs. 7 and 8. This case corresponds to  $n = 6$ ,  $q = 1$ , and the data is fixed to be 000110.  $q = 1$  means that we look to find templates that match at least the last five qubits, i.e., 000110 and 000111. This is the same scenario as the analytical example we presented in Fig. 2, and described in Sec. IID. The result of the quantum counting process is shown in Fig. 7, where we can see that the measured values corresponding to the two eigenvalues from Eqs. (22) are the most probable to be obtained. Converting the state indices from binary to decimal, our result is a bimodal distribution with two modes: 2 and 30

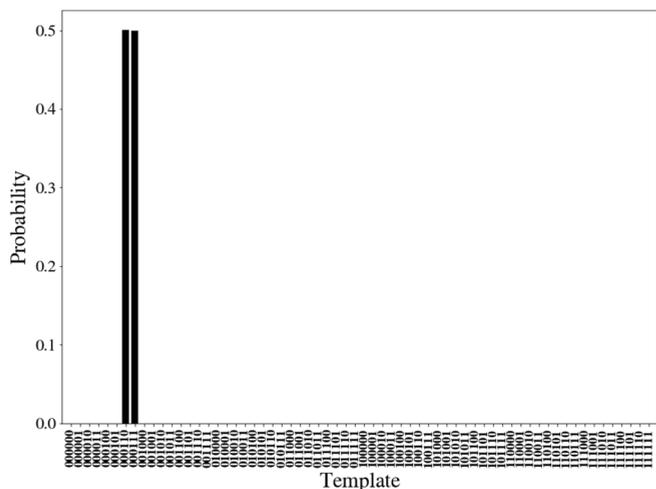


FIG. 8. The measurement of Grover's search process for six-qubit data matching. The data is set as 000110 and the lowest qubit is ignored to allow for two templates matching. With four iterations suggested by the quantum counting process as a numerical output, the two templates that meet the matching criteria are returned with a probability higher than 99% altogether after 2048 trials on the *ibmq\_qasm\_simulator*.

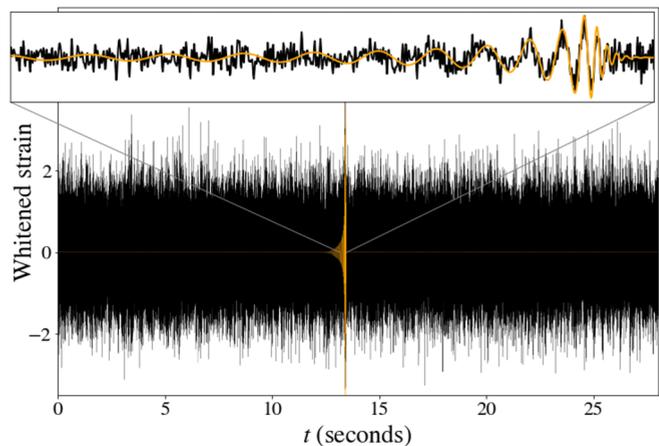


FIG. 9. Whitened time-series data (black) of the gravitational wave event GW150914 sampled at 4096 Hz after a 20 Hz high-pass filter overlaid by a signal template (orange) with component masses  $m_1 = 35.6 M_\odot$  and  $m_2 = 30.6 M_\odot$  and with zero aligned spin, taken from Ref. [87]. The signal can be more clearly seen in the 0.25 s plot in the upper panel.

are the locations of the mode peaks with a standard deviation less than 2. Both cases correspond to an estimate of 4 for  $k_*$ , the same as the true value of  $k$  calculated from the real number of templates. Although this result does not exactly equal that given in Fig. 2, the fact that this algorithm is performed on a quantum simulator with a limited number of runs needs to be taken into consideration.

In Fig. 8, we show the result of Grover's search process based on the result from Fig. 7, in which the two matching templates are recovered with high probability in relation to other templates. Since they form an equal superposition, the two matched templates are assigned approximately equal probability. After performing 2048 trials of simulation in our results, the two matched templates altogether constitute a success probability  $>99\%$ .

## V. EXAMPLE SEARCH FOR GW150914

We now consider how this method can be used in the context of GW astronomy, namely, the detection of the first GW event GW150914 [42]. In this more complex scenario, the data and template bank sizes are too large to be analyzed using IBM's Qiskit library, but we can compute the amplitudes of quantum states that correspond to the template and counting register at various stages of the algorithm described in Sec. III. This is carried out on Python code that is publicly available on Github [86]. The gravitational wave strain time-series data that we choose to analyze is from the LIGO Hanford detector and is centered around the GW150914 event time (GPS time 1126259462.4). It is 28 s in duration and sampled at a rate of 4096 Hz. The data is initially whitened and passed through a high-pass filter with a 20 Hz lower cutoff frequency. The resulting time series is shown in Fig. 9 in black. An approximate matching template is plotted overlaying the data in orange. We perform our analysis on a bank of  $2^{17}$  templates covering the four-dimensional search space defined by the component masses  $m_{1,2}$  and the aligned spin magnitudes  $s_{1,2}$  of the

binary system. We search these templates to find instances that correspond to matching templates.

We first consider applying the SIGNAL DETECTION procedure of Algorithm 2 to determine if a signal is present in the data, and acquire an estimate on the number of matching templates in Sec. V A. In Sec. V B, we show how to continue the analysis by using the TEMPLATE RETRIEVAL procedure of Algorithm 3 to obtain matching templates.

### A. Signal detection

First,  $|\psi_0\rangle$  from Eq. (30) is initialized and the strain data is stored in  $|D\rangle$ . The indices for each of the  $N$  templates are represented by  $|i\rangle$  and are put into superposition with the  $2^p$  states in the counting register as described in Algorithm 2 lines 7–9. The controlled Grover’s operator is applied to  $|\psi_0\rangle$  as described by Algorithm 2 lines 10–16 to compare the templates to the data using Algorithm 1 as a subroutine. The templates are created from  $|i\rangle$  to produce  $|T_i\rangle$  as described in lines 3–5 of Algorithm 1. Here this is done by using a lookup table that is computed prior to the analysis [85] that accepts a given index as a key and returns the set of parameters  $\{m_1, m_2, s_1, s_2\}$  corresponding to the template. The parameters are then given to the phenomenological waveform model IMRPHENOMD to produce the template [60,88,89]. For a quantum computer implementation, we anticipate that this step would not be performed using such a lookup table, as this would rely on using qRAM. Instead, an algorithm is required that maps the  $N$  template indices to their respective locations in the parameter space. The details of this algorithm are beyond the scope of this paper but can be based on existing classical algorithms, such as those used for lattice-based template placement [71,81,90–92], as any classical algorithm can be performed on a quantum computer and made reversible with at most polynomial overhead [74].

For each template in the bank, the oracle calculates  $\rho$  for each time step using Eq. (10) and applying the FFT to produce  $\{\rho_i(t_1), \dots, \rho_i(t_M)\}$  where  $M = 28 \times 4096$  is the number of time steps. A classical search algorithm is also written into the oracle to find  $\rho_i^{\max} = \max(\{\rho_i(t_1), \dots, \rho_i(t_M)\})$ . We then simulate the phase kickback as described in Eq. (12), giving  $f(i) = 1$  if template  $i$  is a matching template, corresponding to  $\rho_i^{\max} \geq \rho_{\text{thr}}$  and  $f(i) = 0$  otherwise (nonmatching template). This can be written explicitly as

$$f(i) = \begin{cases} 1 & \text{if } \max\left(\frac{2}{M\Delta t} \left| \text{FFT}\left(\frac{\hat{Q}_{c,i}(f)\hat{h}(f)}{S_n(f)}\right) \right| \right) \geq \rho_{\text{thr}} \\ 0 & \text{else.} \end{cases} \quad (54)$$

On a quantum computer, this function is evaluated for all templates in parallel, but is repeated  $2^p - 1$  times across the counting register. The number of counting qubits is set to  $p = 11$ , which is the fewest number of qubits in the counting register to meet the condition set in Eq. (37). The probability amplitude of states that correspond to matching templates over each of these operations given GW150914 data is illustrated in Fig. 10 for the analysis repeated with  $\rho_{\text{thr}} = 8, 12, 16, 18$ . Over successive iterations the probability amplitude of the states change according to Eq. (20) with  $\theta$  defined in Eq. (15). With larger  $\rho_{\text{thr}}$ , there are fewer matching templates  $r$  and the period of the probability amplitude’s sinusoidal variation over the counting register states consequently increases. As all

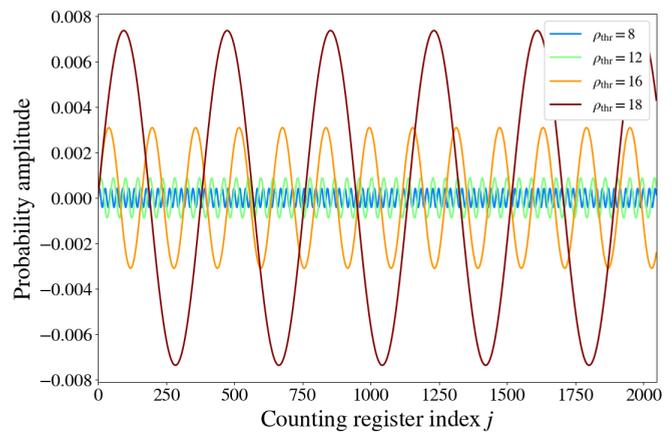


FIG. 10. Probability amplitude of a single matching template over applications of the controlled Grover’s gate specified in lines 10–16 of Algorithm 2 for the four instances of  $\rho_{\text{thr}} = 8, 12, 16, 18$  given GW150914 data and with  $p = 11$ . Larger  $\rho_{\text{thr}}$  decreases the number of matching templates and therefore increases  $k$ . As all matching templates are amplified equally for each case, for a case with fewer matching templates, the total amplitude is divided into fewer equal parts, leading to a larger amplitude for a matching template in comparison to cases with more matching templates.

matching templates are amplified equally with each application, the probability amplitude is divided between fewer states with larger  $\rho_{\text{thr}}$ , which leads to the variations in the amplitude scale seen in Fig. 10.

The amplitudes of the states that correspond to nonmatching templates evolve in a similar sinusoidal fashion as the matching states as shown in Fig. 10 but out of phase. This is illustrated in Fig. 11, where the probability of recovering a matching template  $P(\text{Match})$  (solid line) is compared to

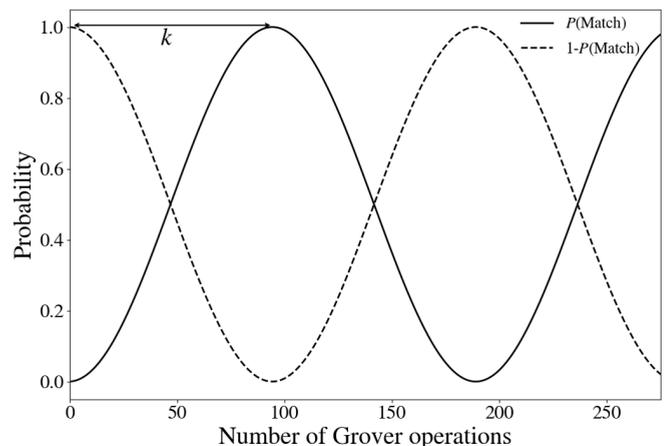


FIG. 11. The probability of returning a matched template (solid line) and nonmatching templates (dashed line) after the template register is measured after  $k$  successive applications of Grover’s operation given the case of  $\rho_{\text{thr}} = 18$  from Fig. 10. The probability amplitude of matching templates follows the sinusoid shown in Fig. 10 for  $\rho_{\text{thr}} = 18$  while that of nonmatching templates follow the same sinusoid but with a  $\pi/2$  phase shift. The probability of returning a matching template is first maximized after  $k$  Grover’s operations.

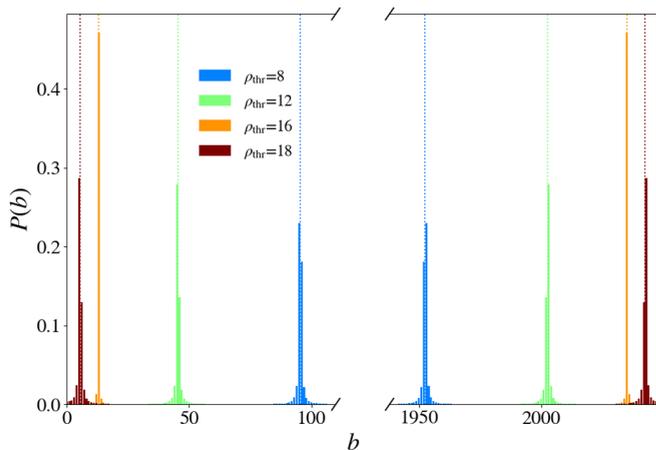


FIG. 12. The probability of different outcomes  $b$  of measuring the counting register after the inverse quantum Fourier transform is applied to the states in Fig. 10. This process is described by lines 17 and 18 for the different cases of  $\rho_{\text{thr}}$  given  $p = 11$ . The distributions are compared to the corresponding value of  $\tilde{b}$  (dotted). The probability distributions corresponding to the two eigenvalues of Grover's operator are closer to  $2^{p-1}$  for cases with more matched templates (lower  $\rho_{\text{thr}}$ ). Cases with fewer matched templates are closer to the extremities of the range of  $b$  and have an increased probability of not identifying any matched templates, corresponding to  $P(b = 0)$ . This probability can be reduced by repeating the algorithm.

the probability of recovering a nonmatching template (dashed line) over successive applications of Grover's operator for the case of  $\rho_{\text{thr}} = 18$ . Initially, all states are equally probable so the probability of returning a matching template is  $r/N$ , and evolve according to Eq. (42) over Grover's operations. The probability of returning a matching template is increased by applying Grover's operator successively until a maximum is reached after  $k$  applications as defined in Eq. (21).

An estimate of the number of matching templates can be made from quantum counting as described in lines 17 and 18 of Algorithm 2 by applying the inverse QFT across the counting register states  $\{|j\rangle\}$  to obtain  $\{|l\rangle\}$ . Figure 12 displays the probabilities of each outcome  $b$  after a measurement is performed on the counting register for the different cases shown in Fig. 10 with  $p = 11$ . The probability of different outcomes after measuring the counting register for the four different cases are compared to the noninteger value  $\tilde{b}$ , defined by the exact solutions of Eqs. (28), and plotted with a dotted line in Fig. 12. The most probable outcome corresponds to  $b'$  or  $b''$  for each case, where the form of the distributions are governed by Eq. (B2). The outcome of measuring the counting register can equally be represented in terms of a prediction of the number of matching templates according to Eqs. (15) and (28) as shown in Fig. 13 for the example cases. For each  $\rho_{\text{thr}}$  considered, the distributions peak near the actual number of matching templates. Notably, the probability of obtaining an outcome that corresponds to a nonzero number of matching templates is much greater than the probability of an outcome corresponding to zero matching templates for all cases. This is equivalent to the probability of obtaining an outcome other than  $b = 0$  in Fig. 12. Obtaining an outcome of  $b = 0$  given the case where there are matching templates

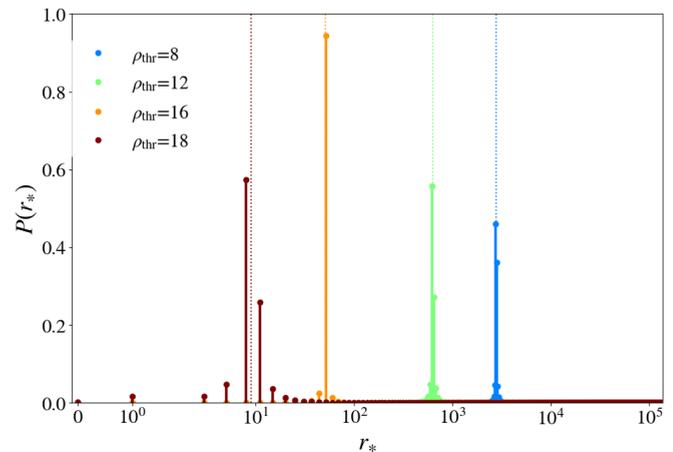


FIG. 13. The probability distributions of outcomes from measuring the counting register from Fig. 12 transformed to estimates on the number of matching templates  $r_*$  for each of the different cases of  $\rho_{\text{thr}}$ . The distributions are compared to the true number of matching templates  $r$  (dotted).

is a false negative, the probability of which is governed by Eq. (40). Therefore, the rate of false negatives (made in addition to that produced from the classical matched filtering approach) can be reduced by repeating the SIGNAL DETECTION procedure. This should be compared to the case where there are no matching templates to identify. In this case, the measurement of the counting register always results in  $b = 0$  corresponding to no matching templates. This negates the possibility of the analysis producing additional false alarms to the classical matched filtering approach as  $P(r_* > 0 | r = 0) = 0$ . If we only wish to determine if a signal is present in the data or not, then the analysis can stop at this stage after the counting register is measured. The cost of determining this outcome requires  $2^p - 1$  enquiries of the oracle, in comparison to the  $\sim O(N)$  calculations of  $\{\rho(t_1), \dots, \rho(t_M)\}$  from Eq. (10) in the classical case.

## B. Retrieving matching templates

Similar to how the number of matching templates is estimated from the counting register's measurement outcome in Sec. VA, the optimal number of Grover's operations is estimated using Eq. (21). Figure 14 shows the probability of obtaining different values of  $k_*$  from the measurement for the same cases of  $\rho_{\text{thr}} = 8, 12, 16, 18$  used in the previous section, and shows that the distributions peak around  $k$ , indicated by the dotted line. Figure 14 is truncated at  $(2^{p-1} - 1)/2$ , so as to exclude the outcome corresponding to zero matching templates and only consider outcomes of  $b > 0$ .

Given the resulting  $k_*$ , the TEMPLATE RETRIEVAL procedure in Algorithm 3 can be applied to obtain a matching template. This involves again initializing  $|\psi_0\rangle$  from Eq. (30) and applying GROVER'S GATE in Algorithm 1 to this state iteratively  $k_*$  times. This is done to maximize the probability that measuring the template register will return an index that corresponds to a matching template as illustrated in Fig. 11. Each state that corresponds to a match will be amplified equally so the probability of obtaining any given matching

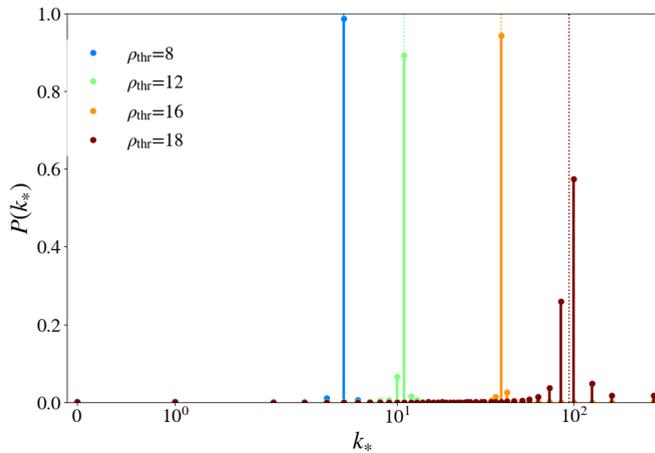


FIG. 14. The probability distributions of outcomes from measuring the counting register from Fig. 12 transformed to estimates on the optimal number of Grover’s applications  $k_*$  for each of the different cases of  $\rho_{\text{thr}}$ . The probabilities are compared to the true  $k$  (dotted) for each case.

template is uniform. For a given  $k_*$ , the probability of obtaining a matching template is governed by Eq. (42).

Figure 15 shows the template states that are amplified from the Grover’s operations in their corresponding positions in the parameter space for each of the different  $\rho_{\text{thr}}$  cases from Sec. V A. The component masses  $m_1$  and  $m_2$  of each binary system are compared to the system’s effective spin  $\chi_{\text{eff}} = (s_1/m_1 + s_2/m_2)/(m_1 + m_2)$ , a reparameterization of the component spins that adequately expresses their effect on the template waveforms in a single parameter. The color of the template markers indicate the maximum  $\rho_{\text{thr}}$  that correspond to them meeting the matching criteria. Note that all templates that correspond to a high  $\rho_{\text{thr}}$  are a subset of lower  $\rho_{\text{thr}}$  values, such that all templates plotted are matches for  $\rho_{\text{thr}} = 8$  but only those marked in red correspond to  $\rho_{\text{thr}} = 18$ . The size of the template labels is scaled to the log probability of obtaining the index of that template from the measurement (where each matching template is obtained with equal probability of  $P(\text{Match})/r$ ), assuming the most probable  $k_*$  Grover’s operations from Fig. 14 are applied. The classically calculated maximum  $\rho$  across all the templates is found to be 19.05 and is highlighted in the figure. This maximum  $\rho$  template coincides with one of the templates that correspond to a match with  $\rho_{\text{thr}} = 18$ .

It must be highlighted that nonoptimal outcomes of measuring the counting register will often occur, and the corresponding  $k_*$  used in the TEMPLATE RETRIEVAL procedure will not maximally amplify the matching template states and increase the probability of failing to retrieve a template as explored in Sec. III C. Even given  $k_* = k$ , there is a nonzero probability of failing to retrieve a template. Therefore, the template that corresponds to the index of a retrieved matching template must be compared to the data using the standard classical method to confirm that a true matching template has been retrieved, by comparing the resulting  $\rho$  to  $\rho_{\text{thr}}$ . If a matching template was retrieved, then an estimate of the time of arrival and distance can be made by simply fitting the template to the data. If a matching template is not retrieved,

then the TEMPLATE RETRIEVAL algorithm should be repeated. Although repeating the algorithm if a match is not found does not add to the asymptotic complexity of the algorithm, which remains  $O(\sqrt{N})$ , we are also interested in the pre-factors for a rigorous comparison between classical and quantum algorithms. In the remainder of this section, we explore strategies to retrieve a template given a nonzero probability of failure, and benchmark these against the classical case.

If the TEMPLATE RETRIEVAL procedure fails to return a matching template, then we can choose to repeat the algorithm given the same  $k_*$  until a matching template is found. Given the  $\rho_{\text{thr}} = 18$  case with  $p = 11$ , we carry out 10 000 simulations of measuring the counting register after the SIGNAL DETECTION procedure to obtain  $k_*$ , before repeating TEMPLATE RETRIEVAL for each  $k_*$  until a matched template is found. The number of times  $f$  [from Eq. (54)] is evaluated for each simulation is tallied in the red histogram of Fig. 16 with a mean indicated by the red dashed line. This can be compared to the number of times  $f$  is evaluated in the classical search case where the function is called upon for every template, indicated by the black dotted line. An alternative approach after repeated failures to retrieve a matching template may be to assume the given  $k_*$  is suboptimal, and to reapply the SIGNAL DETECTION procedure for another  $k_*$  to use. We caution that as the computational cost of the SIGNAL DETECTION procedure is at least  $\sim 4$  times more costly than TEMPLATE RETRIEVAL the tolerance to the number of failed applications of TEMPLATE RETRIEVAL should be  $\gg 1$ . To illustrate this, a further 10 000 simulations are made as before, but the SIGNAL DETECTION procedure is repeated to give a new  $k_*$  for each application of the TEMPLATE RETRIEVAL procedure, corresponding to a fail tolerance of 1. The number of  $f$  evaluations of these simulations using this extreme method is shown in the blue histogram of Fig. 16 and can be seen to have a much greater cost than the method without a fail tolerance. The intervals between adjacent blue histogram bins correspond to the factors of  $2^p - 1$ , the number of  $f$  evaluations in applying SIGNAL DETECTION to obtain the new  $k_*$ . Interestingly, the mean number of  $f$  evaluations for this extreme case is still significantly less than the classical case of calculating  $f$  for all templates. While some choice of failure tolerance may somewhat reduce the tail of the distribution above  $\sim 2(2^p - 1)$ , this corresponds to a fraction of  $\sim 0.01$  of the simulations when no failure tolerance is applied and is therefore insignificant for this case where  $N/r = 2^{17}/9$  and  $p = 11$ .

For the case when all matching templates are desired, then the step described previously must be repeated as described in step 6 of Sec. III, which leads to matching templates being sampled with replacement. This step would be costly for low  $\rho_{\text{thr}}$  with a large proportion of matching templates, which may occur for a loud signal and a low  $\rho_{\text{thr}}$  used for detection. However, a procedure can be made using these algorithms as subroutines to obtain matches with a high  $\rho$  while searching using a low  $\rho_{\text{thr}}$ ; a low  $\rho_{\text{thr}}$  can initially be assumed for the search specified in Sec. V A, and given a measurement corresponding to  $P(r_* > 0)$ , the value of  $r_*$  obtained can be assessed. If  $r_* \gg 1$ , and the signal is presumed to be loud, then the steps in Sec. V A can be repeated with larger  $\rho_{\text{thr}}$ . This can be repeated to optimize the choice of  $\rho_{\text{thr}}$  until the desired number of templates is obtained. The corresponding

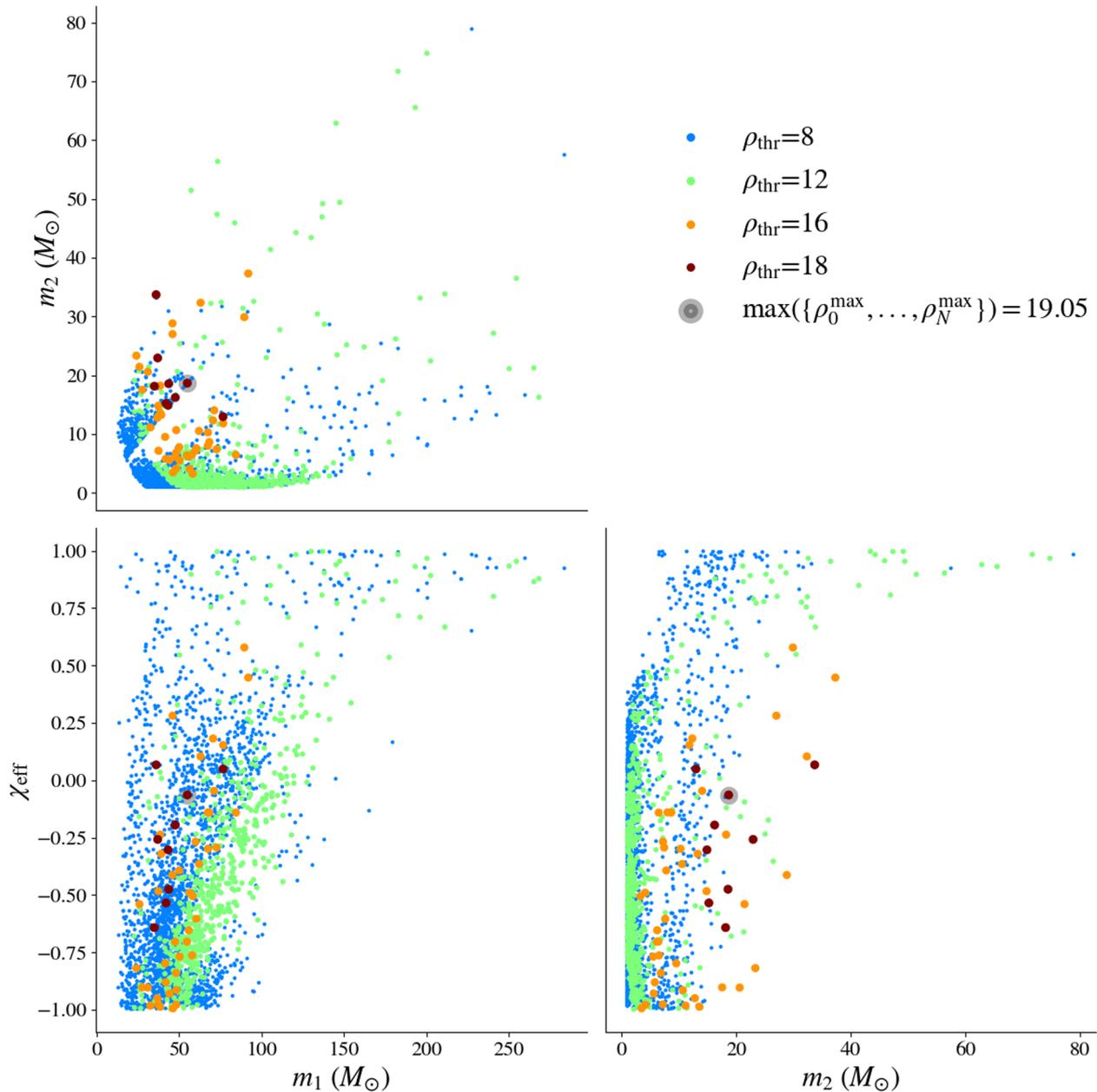


FIG. 15. The positions of templates in the bank that have their corresponding states amplified after applying Grover's operator  $k_*$  times to an initially equal superposition of template states for  $\rho_{\text{thr}}$ . Here  $k_*$  is assumed to be the most probable  $k_*$  from the outcome probabilities shown in Fig. 14. The templates are scattered across the binary system's component masses  $m_1$  and  $m_2$  as well as the effective spin  $\chi_{\text{eff}}$ . The template marker size is proportional to the log probability of obtaining that template state from a measurement of the template register. With increasing  $\rho_{\text{thr}}$ , the matching templates cluster more tightly together and around the template found to have the maximum  $\rho$  out of all the template (found from a classical search).

value of  $k_*$  from this step can then be used to amplify the matching templates. However, each step of this optimization approach requires applying the more computationally costly SIGNAL DETECTION procedure and therefore should be made as to minimize the number of steps, which is a point for future work.

## VI. APPLICATION: CONTINUOUS WAVES

The toy model example using Qiskit and the realistic practical example applied to the GW150914 data serve primarily

as demonstrations of the method. The most impactful application of this algorithm for gravitational-wave (GW) data analysis is for problems where the optimal matched filtering approach is intractable via current classical computing. The continuous GW case is such a problem due to the vast numbers of templates required to cover the search space for unknown continuous wave sources to perform a fully coherent search. A fully coherent search is one in which the match between template and data assumes phase coherence for the duration of the data span. Semicohherent approaches use shorter data segments, requiring significantly less templates,

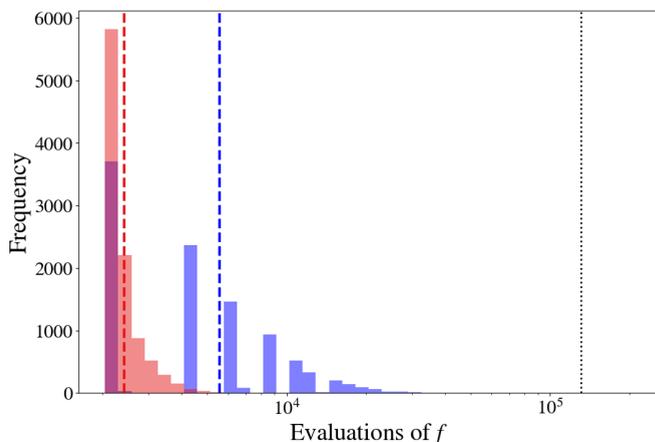


FIG. 16. The number of evaluations of  $f$  required to retrieve a matching template for 10000 simulations given the GW150914 example with  $\rho_{\text{thr}} = 18$  and  $p = 11$ . The red histogram corresponds to simulations where the value of  $k_*$  from the SIGNAL DETECTION procedure is assumed for TEMPLATE RETRIEVAL, which is repeated until a matching template is found. The blue histogram depicts simulations where the quantum counting algorithm is repeated to obtain a new  $k_*$  for each application of the TEMPLATE RETRIEVAL algorithm. The mean for both extreme methods of  $\sim 2,418$  and  $\sim 5,575$  (red, blue dashed lines respectively) are compared to the classical case where all  $2^{17}$  templates are evaluated (dotted line).

and then incoherently combine results from each segment. This latter approach is computationally feasible but has reduced sensitivity.

If performing a fully coherent search for a continuous GW signal, the simplest model to assume for the time-varying signal phase as defined at the solar system barycenter (SSB) can be further expressed as the Taylor expansion

$$\Phi(t_{\text{SSB}}, \vec{\theta}) = \phi_0 + 2\pi \sum_{k=1} \frac{f_k t_{\text{SSB}}^k}{k!}, \quad (55)$$

where  $f_k$  is the  $k$ th derivative of the phase with respect to the solar system barycentre (SSB) time. We further require the transformation between the times defined at the (SSB) and the detector frame which we represent as

$$t_{\text{SSB}} = t + \vec{r}(t) \cdot n(\alpha, \delta_d) + \delta t_{\text{parallax}} + \delta t_{\text{Shapiro}} + \delta t_{\text{Einstein}} + \delta t_{\text{binary}}. \quad (56)$$

The first term here (the Roemer delay) is the dominating contribution to the timing correction. This term is due to the varying position of the detector  $\vec{r}(t)$  as the Earth spins and the orbits the Sun relative to the position of the source on the sky. We denoted the source position by the unit vector  $n(\alpha, \delta_d)$  dependent on the right ascension  $\alpha$  and declination  $\delta_d$ . For observations of length  $\sim 1$  year, it is orbital motion, in particular, that then dictates the number and density of templates that are required on the sky parameters.

A rigorous calculation of the parameter-space metric governing the sky and the GW frequency and its derivatives can be found in Ref. [70] when applied to the so-called  $\mathcal{F}$ -statistic [62]. This statistic is the maximum likelihood ratio for a given template location analytically maximized over the four

amplitude parameters (the received strain amplitude  $h_0$ , the initial reference phase  $\phi_0$ , the polarization angle  $\psi$ , and the inclination angle  $\iota$ ) governing a continuous signal. A useful approximation of the number of required templates can be obtained by considering the allowed variation in each of the search parameters that would lead to a 1 radian phase difference over the course of an observation. This is based on the fact that such a phase difference between signal and template would result in a tolerable level of SNR loss for a coherent analysis. This order of magnitude calculation gives us

$$N \sim 2 \times 10^{28} \left( \frac{f}{1\text{kHz}} \right)^2 \left( \frac{T}{1\text{year}} \right)^3 \left( \frac{\Delta f}{1\text{Hz}} \right) \left( \frac{\Delta f_1}{10^{-9}\text{Hz s}^{-1}} \right) \quad (57)$$

as the total number of templates to search the entire sky over a 1Hz frequency band at 1kHz. Typical searches are performed on small subbands analyzed in parallel on  $\sim 1000$  node computing clusters.

In a similar fashion to the technique used in the compact binary coalescence (CBC) search to optimize the search over time of arrival, the FFT can be used to optimize the search over the intrinsic frequency  $f_0$ . Hence the template bank can be divided into the Cartesian product between frequency templates and the remainder, where the overall classical search cost is linear in the number of templates over the sky and frequency derivative,

$$N_{\text{sky}, f_1} \sim 10^{20} \left( \frac{f}{1\text{kHz}} \right)^2 \left( \frac{T}{1\text{year}} \right)^2 \left( \frac{\Delta f_1}{10^{-9}\text{Hz s}^{-1}} \right), \quad (58)$$

but the joint cost of calculating the detection statistic for a single sky and frequency derivative value, over all possible intrinsic frequencies scales as  $O(N_{f_0} \log N_{f_0})$ , where

$$N_{f_0} \sim 2 \times 10^8 \left( \frac{T}{1\text{year}} \right). \quad (59)$$

The total number of templates in this simple scenario, even when considering a narrow band 1Hz search is many orders of magnitude greater than the number searched in previous analyses (in Ref. [22], the total number of templates searched was  $\sim 10^{14}$  which also included templates over the second frequency derivative  $f_2$ ). Hence, the fully coherent all-sky search over frequency and frequency derivative for one year of data is currently completely infeasible using classical computing.

We have shown that the quantum approach offers a speedup of  $O(\sqrt{N_{\text{sky}, f_1}})$  in the number of calculations required. However, the big  $O$  notation refers to asymptotic scaling and tells us nothing about the prefactors, which could be different in the classical and quantum cases. To claim an expected improvement for a particular case, we need to be a bit more precise. To be specific, for the calculation of the detection statistic, the quantum algorithm requires precisely the same steps as the classical algorithm but requires these to be done in a reversible way, and in addition requires the reversal of the calculation to be performed each time to disentangle these registers from the index register. Standard techniques may be used to construct reversible versions of classical Boolean circuits, which may be implemented directly as quantum circuits. Any classical circuit with  $T$  gates and  $S$  bits may be converted to a reversible circuit with  $O(T^{1+\Delta})$  gates and  $O(S \log T)$  bits. Specifically, for any  $\Delta > 0$  it is possible to construct a reversible circuit

in which the number of gates required is upper bounded by  $3T^{1+\Delta}$  [72]. We thus neglect the factor  $T^\Delta$ , which may be made arbitrarily small, leading to a factor of 3 in the number of gates required. The requirement to erase the intermediate calculations adds a further factor of 2, thus there is a factor of 6 in the number of gates required for the detection statistic calculation in the quantum algorithm compared to the classical algorithm.

Classically, to be certain there is no signal, we need to check against all templates, so we require  $N_{\text{sky},f_1}$  such calculations. In the quantum algorithm to determine whether there is a match or not, we choose  $p$  to be the smallest integer larger than  $p = \log_2(\pi\sqrt{N})$ , requiring around  $\pi \times 10^{10}$  iterations. This gives a false negative with probability at most  $1/\pi^2$ . As discussed,  $\ell$  repetitions of the whole procedure reduce the probability of a false negative to  $\pi^{-2\ell}$ . Thus, e.g., a false negative probability of order  $10^{-6}$  requires six repetitions. Finally, the inverse Fourier transform and measurement steps result in an addition of a logarithmic number of gates and may be neglected. Overall, for a false negative probability of  $10^{-6}$ , we therefore require around  $2 \times 10^{11}$  iterations, each of which requires a factor of 6 more gates than the classical calculation. The overall number of gates needed is of order  $10^{12}T$ , compared to  $10^{20}T$  classically, representing a reduction by a factor of  $10^8$  in the number of operations required.

## VII. DISCUSSION

We have presented a quantum algorithm for matched filtering for GW data analysis. Our algorithm, based on Grover's search algorithm, offers a square-root speedup in the computational cost of searching through a large template bank. As the number of templates is the limiting factor regarding computational feasibility in GW analysis for certain astrophysical signals, this is a natural application of Grover's algorithm. The key theoretical insight that we have used is that for problems of astrophysical interest, the templates are readily computable from theoretical models, and need not be prestored in a database, thus eliminating the need for qRAM. This allows us to construct an oracle, which is readily used in Grover's algorithm, and its extension in quantum counting, to determine whether there are templates that produce an SNR above a given threshold, and to find matching templates.

We have presented proof-of-principle demonstrations of template matching on IBM Qiskit, and through a Python simulation applied to actual GW data. We have also discussed the application to continuous wave searches, currently infeasible with classical techniques. We have focused on applications to GW data analysis, but the algorithm presented here could of course be readily applied to any template matching problem in which the number of templates is much bigger than the size of any one template, and in which the templates can be calculated efficiently.

As we are still some way from scalable, error-corrected quantum processors, it is worth outlining the space requirements of our algorithm, as well as the gate complexity. With  $N$  templates and signal data consisting of  $M$  time steps, we require a counting register of size  $\lceil \log_2 \pi + \frac{1}{2} \log_2 N \rceil$  qubits, an index register of  $\log_2 N$  qubits, and two registers of  $64M$  qubits (if each time sample is stored as a floating

point number, using 8 bytes, or 64 classical bits): one to store the data, and one to store a template corresponding to each index. Recall that these are stored in superposition, so only one template register is needed. In addition, to produce the templates and perform the matched filtering calculation reversibly, we introduce a modest space overhead logarithmic in  $M$ . The dominant contribution to the overall space needed is therefore the size of the data. For the example given in Sec. V, this is 28 seconds of data at 4096 Hz, giving  $M = 28 \times 4096$ . With 8 bytes for each data point, our algorithm becomes feasible with an error-corrected device with a few Megabytes of memory. For fully coherent searches over longer data sets, this increases linearly, and the continuous wave application discussed in section VI requires around 3Gb of memory. The current state of the art is around 50–100 physical qubits [3,93]. Nonetheless, IBM's ambitious quantum hardware road map aims for over 1000 qubits by 2023, in their proposed Condor processor, a device that they view as "a milestone that marks our ability to implement error correction and scale up our devices" [93].

We note also that we have discussed so far only the gate complexity. In the first error-corrected devices, quantum gate operations will be much slower than their classical counterparts, due to both intrinsic gate operation times and the overhead introduced by quantum error-correction. Quadratic speedups, such as that discussed here, do not seem to be promising for runtime advantages for modest fault-tolerant devices [94]. Taking this into account, combined with the quite demanding space requirements outlined above, we do not claim this as a near-term application. However, in the medium to long term with improvements in quantum hardware and in error correction, quantum algorithms have the potential to offer significantly improved sensitivity in GW searches.

This represents just the first step in constructing possible applications of quantum computation to GW data analysis. Employing Grover's algorithm to speed up the search for a match within a large template bank is the first natural step in exploring connections between the two fields. Possibilities for improvement could be to incorporate prior knowledge into the initial state prepared, giving higher weighting to templates considered *a priori* more likely. This has already been explored classically [95,96], and as long as the resulting superposition may be prepared efficiently, such approaches remain amenable to amplitude amplification [97]. The speedup relative to the classical case would remain quadratic but the overall efficiency of both algorithms can be improved.

Another possibility is employing amplitude encoding to store the data and templates. In amplitude encoding, the amplitude of the data at a given time point is stored as the amplitude of a quantum state. This would significantly reduce the space requirements from an  $O(M)$  qubit processor to  $O(\log M)$ . The advantage of the digital encoding we have used here is that arithmetic operations needed to produce the templates and compute the SNR to check for a match above threshold are readily translated from classical circuits. The required matching is more challenging using amplitude encoding, and would likely add to the complexity of this step. A final possibility is to apply machine-learning techniques, either in digital or amplitude encoding, to analyze GW data. This seems promising as machine learning is considered a

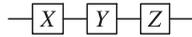


FIG. 17. The Pauli gates expressed in a quantum circuit.

promising area of study for applications for NISQ devices [25]. Classical machine-learning techniques are beginning to be employed in GW detection [98,99] as well as other GW areas [100], and we expect that more sophisticated quantum machine learning techniques may yield further quantum advantages. Exploring the possibility of amplitude encoding, and of quantum machine learning are, however, left for future work. As we fully enter the era of GW astronomy, better performing and more efficient data processing techniques will be needed to fully exploit this new window on the universe. In parallel, as we embark on an era of quantum computational advantage, we anticipate a fruitful interplay between the two fields in harnessing the new computational capabilities offered by this emerging technology.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Science and Technology Facilities Council of the United Kingdom. C.M. and J.V. are supported by the Science and Technology Facilities Council (Grant No. ST/ L000946/1). F.H. is supported by STFC Grant No. ST/N504075/1. S.C. is supported by a Leverhulme fellowship (RF-2020-397).

APPENDIX A: QUANTUM GATES

A quantum computer is roughly composed of three parts: (1) quantum registers to store qubits, (2) a series of quantum gates to perform unitary transformations on the input states, and (3) the measurement procedure to read out the final result.

The qubits have only two orthogonal states, similar to classical computation. The computational basis states are labeled by the associated binary string. They are often represented by column vectors as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{A1}$$

The other pair of orthogonal states frequently used are  $|+\rangle$  and  $|-\rangle$ , defined as

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \end{aligned} \tag{A2}$$

Quantum gates are normally represented by unitary matrices. The quantum gates only applied to one qubit are called single-qubit gates and the ones involving multiple qubits are called multiple-qubit gates.

One set of the most frequently used single-qubit gates are the Pauli gates, whose matrix forms are the associated Pauli matrices. They rotate the qubit by  $\pi$  radian around the corresponding axis on the Bloch sphere. The Pauli-X operator is of particular interest because it functions as the classical NOT gate. They are represented in a quantum circuit diagram shown in Fig. 17.



FIG. 18. The Hadamard gate expressed in a quantum circuit.

Another important single qubit gate is the Hadamard gate, which interchanges the states between the computational basis and the  $|+\rangle$  and  $|-\rangle$  basis,

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{A3}$$

and is represented in a quantum circuit as shown in Fig. 18.

The multiple-qubit gates used in this paper are controlled gates, which are often written as  $C^n-U$ . A controlled gate act on the state of two types of qubits: the control qubits and the target qubits. The operation will be applied to the target qubit if and only if all the  $n$  control qubits are in state  $|1\rangle$ . One example would be the CNOT gate,

$$\hat{U}_{\text{CNOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{A4}$$

and its corresponding quantum circuit expression is shown in Fig. 19.

APPENDIX B: PROBABILITY OF FALSE NEGATIVE

Recall the state of the register after inverse Fourier transform  $|\psi_6\rangle$  in Eq. (27). Without losing generality, only one eigenstate is considered for the analysis. According to Eq. (27), the probability of a certain  $|b\rangle$  measured in the whole state would simply be twice of the probability of that in one eigenstate. To consider the amplitude for the measured state  $|b\rangle$  for eigenstate  $|s_+\rangle$ , we can sum up all its amplitude across  $a$ :

$$\begin{aligned} \frac{1}{2^p} \sum_{a=0}^{2^p-1} e^{i2\pi a(\frac{a}{\pi} - \frac{b}{2^p})} |b\rangle &= \frac{1}{2^p} \frac{e^{i2\pi 2^p(\frac{a}{\pi} - \frac{b}{2^p})} - 1}{e^{i2\pi(\frac{a}{\pi} - \frac{b}{2^p})} - 1} |b\rangle \\ &= \frac{1}{2^p} \frac{\sin(\pi 2^p(\frac{\theta}{\pi} - \frac{b}{2^p}))}{\sin(\pi(\frac{\theta}{\pi} - \frac{b}{2^p}))} e^{i\pi(2^p-1)(\frac{a}{\pi} - \frac{b}{2^p})} |b\rangle. \end{aligned} \tag{B1}$$

The probability of state  $|b\rangle$  would be

$$P(b) = \frac{1}{2^{2p}} \left( \frac{\sin(2^p\theta)}{\sin(\theta - \frac{b\pi}{2^p})} \right)^2. \tag{B2}$$

From the discussion in Sec. III A, the only state situation will trigger a no signal result is when  $|b\rangle = 0$ . According to Eq. (B2), the probability of false negative is

$$\begin{aligned} P(r_* = 0 | r > 0) &= P(b = 0) \\ &= \frac{1}{2^{2p}} \left( \frac{\sin(2^p\theta)}{\sin(\theta)} \right)^2. \end{aligned} \tag{B3}$$



FIG. 19. The CNOT gate expressed in a quantum circuit.

- [1] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, *Rev. Mod. Phys.* **92**, 015003 (2020).
- [2] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, *Chem. Rev.* **120**, 12685 (2020).
- [3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature (London)* **574**, 505 (2019).
- [4] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu *et al.*, Quantum computational advantage using photons, *Science* **370**, 1460 (2020).
- [5] A. Montanaro, Quantum algorithms: An overview, *npj Quantum Inf.* **2**, 15023 (2016).
- [6] R. Orus, S. Mugel, and E. Lizaso, Quantum computing for finance: Overview and prospects, *Rev. Phys.* **4**, 100028 (2019).
- [7] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, *Rev. Mod. Phys.* **86**, 153 (2014).
- [8] A. Blance and M. Spannowsky, Quantum machine learning for particle physics using a variational quantum classifier, *J. High Energy Phys.* **02** (2021) 212.
- [9] D. Magano, A. Kumar, M. Kālis, A. Locāns, A. Glos, S. Pratapsi, G. Quinta, M. Dimitrijevs, A. Rivošs, P. Bargassa *et al.*, Quantum speedup for track reconstruction in particle accelerators, [arXiv:2104.11583](https://arxiv.org/abs/2104.11583) (2021).
- [10] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, *Rep. Prog. Phys.* **81**, 074001 (2018).
- [11] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature (London)* **549**, 195 (2017).
- [12] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing Pages* (Association for Computing Machinery, Philadelphia, Pennsylvania, 1996), pp. 212–219.
- [13] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Sci. Statist. Comput.* **26**, 1484 (1997).
- [14] C. Durr and P. Hoyer, A quantum algorithm for finding the minimum, [arXiv:quant-ph/9607014](https://arxiv.org/abs/quant-ph/9607014).
- [15] E. Aïmeur, G. Brassard, and S. Gambs, Quantum speed-up for unsupervised learning, *Mach. Learn.* **90**, 261 (2013).
- [16] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum nearest-neighbor algorithms for machine learning, *Quantum Inf. Comput.* **15**, 318 (2015).
- [17] H. Ramesh and V. Vinay, String matching in  $O(n+m)$  quantum time, *J. Discrete Algorithms* **1**, 103 (2003).
- [18] A. Montanaro, Quantum pattern matching fast on average, *Algorithmica* **77**, 16 (2017).
- [19] P. Niroula and Y. Nam, A quantum algorithm for string matching, *npj Quantum Inf.* **7**, 37 (2021).
- [20] C. W. Helstrom, *Statistical Theory of Signal Detection* (Elsevier, Exeter, Great Britain, 1968).
- [21] A. C. Searle, Monte-Carlo and Bayesian techniques in gravitational wave burst data analysis, [arXiv:0804.1161](https://arxiv.org/abs/0804.1161).
- [22] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt *et al.*, Searches for continuous gravitational waves from 15 supernova remnants and Fomalhaut b with Advanced LIGO, *Astrophys. J.* **875**, 122 (2019).
- [23] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [24] S. Aaronson, Read the fine print, *Nat. Phys.* **11**, 291 (2015).
- [25] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [26] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, *Proc. R. Soc. London Ser. A* **474**, 20170551 (2018).
- [27] E. Tang, Quantum Principal Component Analysis only Achieves an Exponential Speedup Because of its State Preparation Assumptions, *Phys. Rev. Lett.* **127**, 060503 (2021).
- [28] R. Schützhold, Pattern recognition on a quantum computer, *Phys. Rev. A* **67**, 062311 (2003).
- [29] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, [arXiv:1307.0411](https://arxiv.org/abs/1307.0411).
- [30] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, *Nat. Phys.* **10**, 631 (2014).
- [31] M. Schuld, I. Sinayskiy, and F. Petruccione, Quantum computing for pattern classification, in *Pacific Rim International Conference on Artificial Intelligence* (Springer International Publishing, Cham, Switzerland, 2014), pp. 208–220.
- [32] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, *Phys. Rev. Lett.* **109**, 050505 (2012).
- [33] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum Boltzmann Machine, *Phys. Rev. X* **8**, 021050 (2018).
- [34] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Saarbrücken/Wadern, Germany, 2019).
- [35] M. Sasaki, A. Carlini, and R. Jozsa, Quantum template matching, *Phys. Rev. A* **64**, 022317 (2001).
- [36] M. Sasaki and A. Carlini, Quantum learning and universal quantum matching machine, *Phys. Rev. A* **66**, 022303 (2002).
- [37] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum Fingerprinting, *Phys. Rev. Lett.* **87**, 167902 (2001).
- [38] M. Fanizza, M. Rosati, M. Skotiniotis, J. Calsamiglia, and V. Giovannetti, Beyond the Swap Test: Optimal Estimation of Quantum State Overlap, *Phys. Rev. Lett.* **124**, 060503 (2020).
- [39] J. Preskill, Fault-tolerant quantum computation, *Introduction to Quantum Computation and Information* (World Scientific, Singapore, 1998), pp. 213–269.
- [40] D. Gottesman, An introduction to quantum error correction and fault-tolerant quantum computation, *Quantum Information Science and its Contributions to Mathematics, Proceedings of Symposia in Applied Mathematics*, Vol. 68 (American Mathematical Society, Washington, DC, 2010), pp. 13–58.
- [41] M. S. Anis, H. Abraham, R. A. AduOffei, G. Agliardi, M. Aharoni, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, S. Anagolum *et al.*, Qiskit: An open-source framework for quantum computing, Zenodo Version: 0.7.2 (2019), doi:[10.5281/zenodo.2562111](https://doi.org/10.5281/zenodo.2562111).

- [42] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari *et al.*, Observation of Gravitational Waves from a Binary Black Hole Merger, *Phys. Rev. Lett.* **116**, 061102 (2016).
- [43] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt *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).
- [44] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya *et al.*, GW170817: Generaliz of Neutron Star Radii and Equation of State, *Phys. Rev. Lett.* **121**, 161101 (2018).
- [45] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt *et al.*, Tests of general relativity with binary black holes from the second LIGO-Virgo gravitational-wave transient catalog, *Phys. Rev. D* **103**, 122002 (2021).
- [46] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya *et al.*, Gravitational waves and gamma-rays from a binary neutron star merger: GW170817 and GRB 170817A, *Astrophys. J. Lett.* **848**, L13 (2017).
- [47] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt *et al.*, Population properties of compact objects from the second LIGO-Virgo gravitational-wave transient catalog, *Astrophys. J. Lett.* **913**, L7 (2021).
- [48] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt *et al.*, Gravitational-wave measurement of the Hubble constant following the second observing run of advanced LIGO and Virgo, *Astrophys. J.* **909**, 218 (2021).
- [49] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya, C. Affeldt, M. Agathos *et al.*, Prospects for observing and localizing gravitational-wave transients with advanced LIGO, Advanced Virgo and KAGRA, *Living Rev. Rel.* **21**, 3 (2018).
- [50] J. Aasi, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari *et al.* (The LIGO Scientific Collaboration), Advanced LIGO, *Class. Quantum Grav.* **32**, 074001 (2015).
- [51] F. Acernese, M. Agathos, K. Agatsuma, D. Aisa, N. Allemandou, A. Allocca, J. Amarni, P. Astone, G. Balestri, G. Ballardin *et al.*, Advanced Virgo: A second-generation interferometric gravitational wave detector, *Class. Quantum Grav.* **32**, 024001 (2015).
- [52] T. Akutsu, M. Ando, K. Arai, Y. Arai, S. Araki, A. Araya, N. Aritomi, Y. Aso, S.-W. Bae, Y.-B. Bae *et al.*, Overview of KAGRA: Detector design and construction history, *Prog. Theor. Exper. Phys.* **2021**, 05A101 (2021).
- [53] LIGO-India, Proposal of the Consortium for Indian Initiative in Gravitational-Wave Observations (IndIGO), <https://dcc.ligo.org/LIGO-M1100296/public>, accessed Nov. 19, 2020.
- [54] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt *et al.*, Gravitational-wave constraints on the equatorial ellipticity of millisecond pulsars, *Astrophys. J. Lett.* **902**, L21 (2020).
- [55] R. Balasubramanian, B. S. Sathyaprakash, and S. V. Dhurandhar, Gravitational waves from coalescing binaries: Detection strategies and Monte Carlo estimation of parameters, *Phys. Rev. D* **53**, 3033 (1996).
- [56] B. J. Owen, Search templates for gravitational waves from inspiraling binaries: Choice of template spacing, *Phys. Rev. D* **53**, 6749 (1996).
- [57] 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).
- [58] P. R. Brady, T. Creighton, C. Cutler, and B. F. Schutz, Searching for periodic sources with LIGO, *Phys. Rev. D* **57**, 2101 (1998).
- [59] 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).
- [60] S. Husa, S. Khan, M. Hannam, M. Pürrer, F. Ohme, X. J. Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. I. New numerical waveforms and anatomy of the signal, *Phys. Rev. D* **93**, 044006 (2016).
- [61] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond, T. Chu, H. Fong, P. Kumar, H. P. Pfeiffer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, M. A. Scheel, and B. Szilágyi, Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors, *Phys. Rev. D* **95**, 044028 (2017).
- [62] P. Janowski, A. Królak, and B. F. Schutz, Data analysis of gravitational-wave signals from spinning neutron stars: The signal and its detection, *Phys. Rev. D* **58**, 063001 (1998).
- [63] S. V. Dhurandhar and A. Vecchio, Searching for continuous gravitational wave sources in binary systems, *Phys. Rev. D* **63**, 122001 (2001).
- [64] C. Messenger, Semicoherent search strategy for known continuous wave sources in binary systems, *Phys. Rev. D* **84**, 083003 (2011).
- [65] P. Leaci and R. Prix, Directed searches for continuous gravitational waves from binary systems: Parameter-space metrics and optimal Scorpius X-1 sensitivity, *Phys. Rev. D* **91**, 102003 (2015).
- [66] 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).
- [67] D. Macleod, I. W. Harry, and S. Fairhurst, Fully-coherent all-sky search for gravitational-waves from compact binary coalescences, *Phys. Rev. D* **93**, 064004 (2016).
- [68] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, Historical notes on the fast Fourier transform, *Proc. IEEE* **55**, 1675 (1967).
- [69] 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).
- [70] R. Prix, Search for continuous gravitational waves: Metric of the multidetector F-statistic, *Phys. Rev. D* **75**, 023004 (2007).
- [71] C. Messenger, R. Prix, and M. A. Papa, Random template banks and relaxed lattice coverings, *Phys. Rev. D* **79**, 104017 (2009).

- [72] E. Rieffel and W. Polak, *Quantum Computing: A Gentle Introduction* (MIT Press, 2011).
- [73] S. Barnett, *Quantum Information* (Oxford University Press, New York, 2009), Vol. 16.
- [74] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and weaknesses of quantum computing, *SIAM J. Comput.* **26**, 1510 (1997).
- [75] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, New York, 2010).
- [76] P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing* (Oxford University Press, New York, 2007).
- [77] P. Flajolet, D. Gardy, and L. Thimonier, Birthday paradox, coupon collectors, caching algorithms and self-organizing search, *Discrete Appl. Math.* **39**, 207 (1992).
- [78] G. Brassard, P. Høyer, and A. Tapp, Quantum counting, in *International Colloquium on Automata, Languages, and Programming* (Springer, Berlin, Heidelberg, 1998), pp. 820–831.
- [79] M. Mosca, Counting by quantum eigenvalue estimation, *Theor. Comput. Sci.* **264**, 139 (2001).
- [80] A. Yu. Kitaev, Quantum computations: Algorithms and error correction, *Russ. Math. Surveys* **52**, 1191 (1997).
- [81] R. Prix, Template-based searches for gravitational waves: Efficient lattice covering of flat parameter spaces, *Class. Quantum Gravity* **24**, S481 (2007).
- [82] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [83] J. Veitch, V. Raymond, B. Farr, W. Farr, P. Graff, S. Vitale, B. Aylott, K. Blackburn, N. Christensen, M. Coughlin, W. Del Pozzo, F. Feroz, J. Gair, C. J. Haster, V. Kalogera, T. Littenberg, I. Mandel, R. O’Shaughnessy, M. Pitkin, C. Rodriguez, C. Röver, T. Sidery, R. Smith, M. Van Der Sluys, A. Vecchio, W. Vousden, and L. Wade, Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library, *Phys. Rev. D* **91**, 042003 (2015).
- [84] G. Ashton, M. Hübner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divarkala, P. J. Easter, B. Goncharov *et al.*, BILBY: A user-friendly Bayesian inference library for gravitational-wave astronomy, *Astrophys. J. Suppl.* **241**, 27 (2019).
- [85] 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](https://arxiv.org/abs/1705.01845) (2017).
- [86] <https://github.com/Fergus-Hayes/quantum-matched-filter>.
- [87] B. P. 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).
- [88] 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).
- [89] S. Khan, K. Chatziioannou, M. Hannam, and F. Ohme, Phenomenological model for the gravitational-wave signal from precessing binary black holes with two-spin effects, *Phys. Rev. D* **100**, 024059 (2019).
- [90] T. Cokelaer, Gravitational waves from inspiralling compact binaries: Hexagonal template placement and its efficiency in detecting physical signals, *Phys. Rev. D* **76**, 102004 (2007).
- [91] 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).
- [92] 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).
- [93] IBM’s roadmap for scaling quantum technology, <https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/>, accessed July 14, 2021
- [94] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, Focus beyond quadratic speedups for error-corrected quantum advantage, *PRX Quantum* **2**, 010103 (2021).
- [95] C. Röver, Random template placement and prior information, *J. Phys.: Conf. Ser.* **228**, 012008 (2010).
- [96] T. Dent and J. Veitch, Optimizing gravitational-wave searches for a population of coalescing binaries: Intrinsic parameters, *Phys. Rev. D* **89**, 062002 (2014).
- [97] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemporary Math.* **305**, 53 (2002).
- [98] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Matching Matched Filtering with Deep Networks for Gravitational-Wave Astronomy, *Phys. Rev. Lett.* **120**, 141103 (2018).
- [99] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith, Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy, *Nature Phys.* **18**, 112 (2022).
- [100] E. Cuoco, J. Powell, M. Cavaglià, K. Ackley, M. Bejger, C. Chatterjee, M. Coughlin, S. Coughlin, P. Easter, R. Essick *et al.*, Enhancing gravitational-wave science with machine learning, *Mach. Learn.: Sci. Technol.* **2**, 011002 (2021).