

# Gravitational wave matched filtering by quantum Monte Carlo integration and quantum amplitude amplification

Koichi Miyamoto,<sup>1,\*</sup> Gonzalo Murrás,<sup>2</sup> Takahiro S. Yamamoto,<sup>3</sup> Sachiko Kuroyanagi,<sup>2,3</sup> and Savvas Nesseris<sup>2</sup>

<sup>1</sup>Center for Quantum Information and Quantum Biology, Osaka University, Toyonaka, 560-8531, Japan

<sup>2</sup>Instituto de Física Teórica UAM-CSIC, Universidad Autónoma de Madrid, Cantoblanco, 28049 Madrid, Spain

<sup>3</sup>Department of Physics and Astrophysics, Nagoya University, Nagoya, 464-8602, Japan

(Dated: May 22, 2022)

The speedup of heavy numerical tasks by quantum computing is now actively investigated in various fields including data analysis in physics and astronomy. In this paper, we propose a new quantum algorithm for matched filtering in gravitational wave (GW) data analysis based on the previous work by Gao et al., Phys. Rev. Research 4, 023006 (2022) [arXiv:2109.01535]. Our approach uses the quantum algorithm for Monte Carlo integration for the signal-to-noise ratio (SNR) calculation instead of the fast Fourier transform used in Gao et al. and searches signal templates with high SNR by quantum amplitude amplification. In this way, we achieve an exponential reduction of the qubit number compared with Gao et al.'s algorithm, keeping a quadratic speedup over classical GW matched filtering with respect to the template number.

As the full version of this paper, see [1].

## I. INTRODUCTION

Quantum computing [2] is a developing technology and is expected to speed up some classes of computation that are intractable in classical computing. The recent rapid advance in quantum computer development has been stimulating research on applications of quantum algorithms to concrete problems in various fields.

In this paper, we study an application of some quantum algorithms to a problem in gravitational wave (GW) experiments, which measures the space-time distortion caused by GWs with a laser interferometer. A recent paper [3] has proposed the use of Grover's search algorithm for matched filtering, which is a commonly used technique to search a signal buried in noisy data and is widely used in GW data analysis [4–7]. Given a target signal waveform, called a template, we take an inner product between a template and data to cancel out the noise contribution and extract a signal. In fact, the first GW was detected by the LIGO detectors in 2015 [8], and, after that, the worldwide GW detector network has observed tens of GW events using matched filtering [9–11].

A challenging point in GW matched filtering is the large number of templates. The functional form of a GW signal is predicted by general relativity depending on the GW source such as compact binary coalescence [4–7], but it has some parameters such as masses of the compact objects, spin parameters, and luminosity distance. In order to get a high signal-to-noise ratio (SNR), we must perform matched filtering using a template with appropriate parameters. Therefore, usually, we set sufficiently many points in the parameter space and run an exhaustive search. That is, we repeat matched filtering using templates one by one to find those that yield SNR larger than some threshold. This is an extremely time-consuming task and expected to be sped up by quantum computing.

Fortunately, there exists a quantum algorithm for searching, called Grover's algorithm [12]. Given  $N$  data  $x_1, \dots, x_N$  represented as bit strings and a condition  $F$  as a function that maps a bit string to 0 or 1, Grover's algorithm can find "marked data"  $x$  such that  $F(x) = 1$  making  $O(\sqrt{N/n})$  calls to  $F$ , where  $n$  is the number of marked data. Therefore, it is often said that Grover's algorithm provides a quadratic speedup over the classical exhaustive search, which has  $O(N/n)$  query complexity. As an application of this, Ref. [3] presented a quantum algorithm for GW matched filtering. With the SNR calculation implemented as a quantum circuit, the aforementioned algorithm can find a template with SNR higher than a threshold  $\rho_{\text{th}}$  with  $\tilde{O}\left(M/\sqrt{r(\rho_{\text{th}})}\right)$  complexity<sup>1</sup>, where  $r(\rho_{\text{th}})$  is the fraction of templates that yield  $\text{SNR } \rho \geq \rho_{\text{th}}$  and  $M$  is the number of points in the time-series data of the detector output, or, equivalently, the number of frequency bins of Fourier transformed data. This is in fact a quadratic speedup over the classical method, which has  $O(M/r(\rho_{\text{th}}))$  complexity, with respect to the template number.

However, this quantum algorithm has the following subtlety. It uses Fast Fourier Transform (FFT) [13] for the SNR calculation, which is also used in the usual classical way. FFT simultaneously calculates SNR for  $M$  possible values of a parameter called time of coalescence, with other parameter fixed, in  $O(M \log M)$  time, whereas naively such a computation takes  $O(M^2)$  time without FFT. However, in order to store the intermediate and final calculation results, FFT requires  $O(M)$  qubits, which is a somewhat large number since  $M$  is typically of order  $4096\text{Hz} \times 256\text{s} \sim O(10^6)$ , where  $4096\text{Hz}$  is the sampling frequency and  $256\text{s}$  is the typical duration of the data segment [7]. This might cause an issue on feasibility, since fault-tolerant quantum computers will have a limitation on the number of qubits available even in the future. It is expected that creating one logical qubit requires thousands or tens of thousands physical qubits for error correction [14],

\* miyamoto.kouichi.qiqb@osaka-u.ac.jp

<sup>1</sup> In the big-O notation, we use a symbol  $\tilde{O}(\cdot)$ , which hides logarithmic factors in  $O(\cdot)$ .

and therefore realizing a quantum computer with millions of logical qubits is very challenging.

In light of this, we propose an alternative quantum algorithm for GW matched filtering, in which the SNR calculation with FFT is replaced with the quantum algorithm for Monte Carlo integration (QMCI) [15]. QMCI is a method to estimate an integral given in the finite sum approximation. Thus, it can be applied to the calculation of SNR, which includes frequency band integration and is in practice calculated as a sum of contributions from many Fourier modes. In this approach, the required qubit number scales on  $M$  as  $O(\text{poly}(\log M))$ , which means an exponential reduction from the FFT approach. Note that this is not just a straightforward application of another quantum algorithm to a part of an existing method, since the use of QMCI causes the following issue. Unlike FFT, which calculates SNR deterministically, QMCI inevitably accompanies errors, and thus comparing the SNR calculated by QMCI with a single SNR threshold  $\rho_{\text{th}}$  leads to a false alarm that the detector output yields SNR larger than  $\rho_{\text{th}}$  for some templates despite there being no such event. As a solution to this, we propose to set two thresholds  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$  that have the following meanings: we should never miss events with SNR  $\rho \geq \rho_{\text{hard}}$ , and we do not want to be falsely alarmed by events with  $\rho < \rho_{\text{soft}}$ . Then, with QMCI accuracy set according to the difference between  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$ , the proposed algorithm says “there is a signal” for events with SNR  $\rho \geq \rho_{\text{hard}}$  with high probability, “there is no signal” for events with SNR  $\rho < \rho_{\text{soft}}$  with certainty, and either of these messages for events with SNR  $\rho \in [\rho_{\text{soft}}, \rho_{\text{hard}}]$ . The query complexity in this algorithm is of order  $\tilde{O}(M/\sqrt{r(\rho_{\text{hard}})})$ , which still indicates a quadratic speedup.

The remaining part of this paper is organized as follows. Section II introduces the preliminary knowledge. We outline GW matched filtering and some building-block quantum algorithms such as Grover’s algorithm, quantum amplitude amplification (QAA), quantum amplitude estimation (QAE), and QMCI. Section III is the main part. Defining GW matched filtering as a mathematical problem, we explain the existing algorithm in [3], and present our modified algorithm in detail, along with estimation of the query complexity and the qubit number and a plausible setting on thresholds  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$ . Section IV summarizes this paper.

## II. PRELIMINARY

### A. Notation

$\mathbb{R}_+$  denotes the set of all positive real numbers. For  $n \in \mathbb{N}$ , we define  $[n] := \{1, \dots, n\}$  and  $[n]_0 := \{0, 1, \dots, n-1\}$ . For any  $x \in \mathbb{R}$ , if  $|x - y| \leq \epsilon$  holds for some  $y \in \mathbb{R}$  and  $\epsilon \in \mathbb{R}_+$ , we say that  $x$  is  $\epsilon$ -close to  $y$  and that  $x$  is an  $\epsilon$ -approximation of  $y$ . For any equation or inequality  $C$ ,  $\mathbb{1}_C$  takes 1 if  $C$  is satisfied, and 0 otherwise. For  $c \in \mathbb{C}$ ,  $c^*$  denotes its complex conjugate. For a finite set of real numbers  $\mathcal{X} = \{x_1, \dots, x_n\}$ , we define the sample mean  $\text{Mean}(\mathcal{X}) := \frac{1}{n} \sum_{i=1}^n x_i$  and the sample variance  $\text{Var}(\mathcal{X}) := \frac{1}{n} \sum_{i=1}^n (x_i - \text{Mean}(\mathcal{X}))^2$ . For  $n \in \mathbb{N}$ ,  $\mathbb{I}_n$  denotes the

$n \times n$  identity matrix. For  $z \in \mathbb{C}$ ,  $\Re z$  and  $\Im z$  are the real and imaginary parts of  $z$ .

### B. Gravitational wave matched filtering

Here, we outline matched filtering in GW search experiments [4–7]. Suppose that we are given the detector output  $s(t)$  as a function of time  $t$ , which is a sum of the signal  $h(t)$  and the noise  $n(t)$ :

$$s(t) = h(t) + n(t). \quad (1)$$

We assume that the noise is Gaussian, which means that, for each  $f \in \mathbb{R}_+$ ,  $\Re \tilde{n}(f)$  and  $\Im \tilde{n}(f)$  are normal random variables and

$$\mathbb{E}_n[\tilde{n}(f)\tilde{n}^*(f')] = \frac{1}{2}S_n(|f|)\delta(f - f') \quad (2)$$

holds with the single-sided power spectrum density (PSD)  $S_n$ . Here, for any function  $q(t)$  in time domain,  $\tilde{q}(f) := \int_{-\infty}^{\infty} dt e^{2\pi i f t} q(t)$  is its Fourier transform,  $\mathbb{E}_n[\cdot]$  denotes an expectation with respect to randomness of the noise and  $\delta(\cdot)$  is the Dirac delta function. We define the inner product of two functions  $q(t)$  and  $q'(t)$  in time domain as

$$(q|q') := 4\Re \left( \int_0^{\infty} df \frac{\tilde{q}^*(f)\tilde{q}'(f)}{S_n(f)} \right). \quad (3)$$

The matched filtering search is performed by taking an inner product of  $s(t)$  and an appropriate filter function  $Q(t)$  that yields a large inner product with the targeted signal. The function  $Q(t)$  is often called a template and normalized as  $(Q|Q) = 1$ . The template bank, the collection of templates, is prepared based on theoretically predicted waveform of signals. The SNR for a detector output  $s$  and template  $Q$  is then defined as

$$\rho = \frac{(Q|s)}{\sqrt{\mathbb{E}_n[(Q|n)]^2}} = 4\Re \left( \int_0^{\infty} df \frac{\tilde{Q}^*(f)\tilde{s}(f)}{S_n(f)} \right). \quad (4)$$

By setting a SNR threshold  $\rho_{\text{th}}$ , exhaustive search is performed to find a template that gives a SNR larger than  $\rho_{\text{th}}$  to claim a detection. Hereafter, we write each Fourier transformed template as  $\tilde{Q}_m(f)e^{-2\pi i f t_0}$ , where  $t_0$  is the time of coalescence and the dependency on other parameters (intrinsic parameters) is put into  $\tilde{Q}_m$ . Here, we assume that there are  $N_{\text{temp}}$  candidates of the intrinsic parameter set in the template bank and label the functions  $\tilde{Q}_m$  by  $m \in [N_{\text{temp}}]_0$ .

In reality, we have a detector output as a sequence of discrete points in time [7]. Suppose that a sequence of  $s(t)$  is given by  $\{s(\tau_l)\}_{l=0, \dots, M-1}$  at  $M$  time points  $\tau_0 = 0, \tau_1 = \Delta t, \dots, \tau_{M-1} = (M-1)\Delta t$  with interval  $\Delta t$ . In such a situation, Fourier transforms are given in the discrete form for each  $k \in \{0, 1, \dots, M-1\}$ . Thus, we redefine  $\tilde{s}$  as

$$\tilde{s}(f_k) := \Delta t \sum_{l=0}^{M-1} s(\tau_l) e^{2\pi i k l / M}, \quad (5)$$

where  $f_k = k/T$  and  $T = M\Delta t$ , and similar quantities  $\tilde{h}(f_k)$  and  $\tilde{n}(f_k)$  for  $h(t)$  and  $n(t)$ .  $\Re\tilde{n}(f_k)$  and  $\Im\tilde{n}(f_k)$  are still normal random variables but Eq. (2) is now converted into [16]

$$\mathbb{E}_n[\tilde{n}(f_k)\tilde{n}^*(f_l)] = \frac{1}{2}S_n(f_k)T\delta_{k,l}, \quad (6)$$

where  $\delta_{k,l}$  is the Kronecker delta. Then, the SNR now becomes

$$\rho_{m,j} = \frac{4}{M\Delta t} \Re \left( \sum_{k=1}^{\frac{M}{2}-1} \frac{\tilde{Q}_m^*(f_k)\tilde{s}(f_k)}{S_n(f_k)} e^{2\pi i j k / M} \right), \quad (7)$$

for the  $m$ -th intrinsic parameter set and the time of coalescence given as  $t_0 = j\Delta t$  with  $j \in [M]_0$ <sup>2</sup>. Here and hereafter, we assume that  $M$  is even. In the usual way of classical computing, although it seemingly takes  $O(M^2)$  computational time to compute Eq. (7) for all  $j \in [M]_0$  with  $m$  fixed, we can do this in  $O(M \log M)$  time using FFT [13]. This means that, for a fixed intrinsic parameter set, we can quickly search the optimal  $t_0$  in  $\{j\Delta t\}_{j=0,\dots,M-1}$  and obtain the optimal SNR  $\rho_m := \max_{j \in [M]_0} \rho_{m,j}$ , which is why  $t_0$  is dealt with separately from other template parameters in the conventional GW data analysis. On the other hand, for the other intrinsic parameters, we calculate  $\rho_m$  for each  $m \in [N_{\text{temp}}]_0$  one by one until we get  $\rho_m \geq \rho_{\text{th}}$ , which results in the number of floating-point operations of order

$$O\left(\frac{M \log M}{r(\rho_{\text{th}})}\right) \quad (8)$$

for  $r(\rho_{\text{th}}) > 0$ . Here

$$r(\rho) := \frac{|\{m \in [N_{\text{temp}}]_0 \mid \rho_m \geq \rho\}|}{N_{\text{temp}}} \quad (9)$$

is the fraction of intrinsic parameter sets in the template banks that yields SNRs larger than  $\rho$ , with time of coalescence optimized. When  $r(\rho_{\text{th}}) = 0$ , we have to go through all the template and thus the number of floating-point operations is

$$O(N_{\text{temp}} M \log M). \quad (10)$$

Unlike the above, as we will see later, we do not use FFT in the proposed quantum method. For later convenience, we rewrite Eq. (7) as

$$\begin{aligned} \rho_{m,j} &= \frac{2}{M} \sum_{k=1}^{\frac{M}{2}-1} \tilde{\rho}_{m,j,k}, \\ \tilde{\rho}_{m,j,k} &:= \Re \left( \frac{2\tilde{Q}_m^*(f_k)\tilde{s}(f_k)}{S_n(f_k)\Delta t} \right) \cos \left( \frac{2\pi j k}{M} \right) \\ &\quad - \Im \left( \frac{2\tilde{Q}_m^*(f_k)\tilde{s}(f_k)}{S_n(f_k)\Delta t} \right) \sin \left( \frac{2\pi j k}{M} \right), \end{aligned} \quad (11)$$

and set  $\tilde{\rho}_{m,j,0} = 0$ .

### C. Representation of real numbers on qubits and some basic quantum circuits

In numerical calculations in this paper, we use bit strings on qubits as fixed-point binary representations of real numbers and, for  $x \in \mathbb{R}$ , we denote by  $|x\rangle$  the computational basis state on a quantum register which corresponds to  $x$ . Unless otherwise stated, we assume that each register has  $N_{\text{dig}}$  qubits, where  $N_{\text{dig}}$  is a sufficiently large positive integer set independently from parameters in problems under consideration, and neglect errors from finite-precision representation.

For computing with real numbers, we use the quantum circuits for four basic arithmetic operations: addition  $O_{\text{add}}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle|x+y\rangle$ , subtraction  $O_{\text{sub}}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle|x-y\rangle$ , multiplication  $O_{\text{mul}}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle|xy\rangle$  and division  $O_{\text{div}}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle|x/y\rangle$ , where  $x, y$  are any real numbers ( $y \neq 0$  for division) and some ancillary registers may be undisplayed. In fact, concrete circuit implementations for these operation have been presented [17–19]. These circuits enable the calculation of rational functions. We also use quantum circuits for calculation of elementary functions  $f$  such as exponential, trigonometric functions, and so on:  $O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$  for any  $x \in \mathbb{R}$ . Such circuits can be implemented through, for example, piecewise polynomial approximation [20].

In addition to these circuits for numerical calculation, we now list some oracles used in the proposed quantum algorithm. A comparer  $O_{\text{comp}}$  acts as  $O_{\text{comp}}|x\rangle|y\rangle|0\rangle = |x\rangle|y\rangle(\mathbb{1}_{x \geq y}|1\rangle + \mathbb{1}_{x < y}|0\rangle)$  for any  $x, y \in \mathbb{R}$ . This is actually equivalent to subtraction  $x - y$ , since, if we adopt 2's complement method to represent negative numbers, the most significant digit represents the sign of a number [21]. A Y-rotation with controlled angle gate  $O_{\text{CY}}$  acts as  $O_{\text{CY}}|\theta\rangle|\psi\rangle = |\theta\rangle \otimes R_Y(\theta)|\psi\rangle$ , where  $R_Y(\theta) := \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$ , for any  $\theta \in \mathbb{R}$  and any single-qubit state  $|\psi\rangle$ . This is implemented as a sequence of fixed-angle controlled Y-rotation gates  $|0\rangle\langle 0| \otimes \mathbb{I}_2 + |1\rangle\langle 1| \otimes R_Y(\theta)$ , where  $\theta \in \mathbb{R}$  is prefixed [22]. We also use the oracle  $O_N^{\text{EqPr}}$ , where  $N \in \mathbb{N}$ , to generate equiprobable superposition of states from  $|0\rangle$  to  $|N-1\rangle$ :  $O_N^{\text{EqPr}}|0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$ . If  $N = 2^n$  with some  $n \in \mathbb{N}$ , we can generate such a state by operating a Hadamard gate on each qubit in a  $n$ -qubit register. Also for  $N$  that is not a power of 2, we can implement  $O_N^{\text{EqPr}}$  by the method in [23] to generate a state in which a given probability density  $p(x)$  is amplitude-encoded, with  $p(x)$  set to the uniform density on  $[0, (N-1)/2^n]$  with  $n = \lceil \log_2 N \rceil$ .

The last oracle we mention here is  $O_N^{\text{med}}$  that, for any  $N$  real numbers  $x_1, \dots, x_N$ , outputs the median  $\text{med}(x_1, \dots, x_N)$ :  $O_N^{\text{med}}|x_1\rangle \cdots |x_N\rangle|0\rangle = |x_1\rangle \cdots |x_N\rangle|\text{med}(x_1, \dots, x_N)\rangle$ . This operation is implemented as follows. First, we transform  $|x_1\rangle \cdots |x_N\rangle$  to  $|x_1^{\text{sort}}\rangle \cdots |x_N^{\text{sort}}\rangle$ , where  $x_1^{\text{sort}}, \dots, x_N^{\text{sort}}$  is a sequence made by ascending sort of  $x_1, \dots, x_N$  and some ancillary qubits are not displayed<sup>3</sup>. Then, we let the number on the

<sup>2</sup> The sum in Eq. (7) runs over  $k \in [\frac{M}{2} - 1]$  rather than  $k \in [M]_0$ , because we use the one-sided power spectral density as defined in Eq. (2). See [7] for the detail.

<sup>3</sup> In this ascending sort operation, unitarity holds in the system including ancillary qubits.

midmost register be  $\text{med}(x_1, \dots, x_N)$ . Note that exchange based sort algorithms such as bubble sort [24] can be implemented since the operation

$$|x\rangle|y\rangle|0\rangle \rightarrow |x\rangle|y\rangle|\mathbb{1}_{x \geq y}\rangle \rightarrow \begin{cases} |y\rangle|x\rangle|1\rangle & ; \text{if } x \geq y \\ |x\rangle|y\rangle|0\rangle & ; \text{otherwise} \end{cases}, \quad (12)$$

is possible for any  $x, y \in \mathbb{R}$ , where we use  $O_{\text{comp}}$  and controlled SWAP gates at the first and second arrows, respectively.

Hereafter, we collectively call the above oracles arithmetic oracles.

#### D. Grover's algorithm and Quantum amplitude amplification

Grover's algorithm [12] is a quantum algorithm for searching a "marked entry"  $x$ , which satisfies some condition given as a binary-valued function, from an unstructured database. Formally, we have the following theorem.

**Theorem 1.** *Let  $n \in \mathbb{N}$  and  $F : \{0, 1\}^n \rightarrow \{0, 1\}$  is a function such that  $F(x_{\text{tar}}) = 1$  for one element  $x_{\text{tar}} \in \{0, 1\}^n$  and  $F(x) = 0$  for any  $x \in \{0, 1\}^n \setminus \{x_{\text{tar}}\}$ . Suppose that we are given an access to an oracle  $O_F$  on a system consisting of a  $n$ -qubit register and a qubit such that  $O_F|x\rangle|0\rangle = O_F|x\rangle|F(x)\rangle$  for any  $x \in \{0, 1\}^n$ . Then, for any  $\delta \in (0, 1)$ , there exists a quantum algorithm that, with probability at least  $1 - \delta$ , outputs  $x_{\text{tar}}$  making  $O(\sqrt{N} \log \delta^{-1})$  calls to  $O_F$ , where  $N := 2^n$ .*

This is often called a quadratic speedup over classical search methods that takes  $O(N)$  time for the same problem.

Besides, there exists a quantum algorithm called QAA [25, 26], which can be seen as an extension of Grover's algorithm. It is an algorithm to amplify the amplitude of the "marked state" in a given superposition and obtain the state. Formally, the following theorem holds.

**Theorem 2.** *Suppose that we are given an access to an oracle  $A$  that acts on a system consisting of a  $n$ -qubit register and a single-qubit register as*

$$A|0\rangle|0\rangle = \sqrt{a}|\phi_1\rangle|1\rangle + \sqrt{1-a}|\phi_0\rangle|0\rangle =: |\Phi\rangle, \quad (13)$$

where  $|\phi_0\rangle$  and  $|\phi_1\rangle$  are some quantum states on the register and  $a \in [0, 1)$ . Then, for any  $\gamma, \delta \in (0, 1)$ , there exists a quantum algorithm QAA( $A, \gamma, \delta$ ) that uses  $O(n)$  qubits and behaves as follows:

- The output of the algorithm is either of
  - (A) the message "success" and the quantum state  $|\phi_1\rangle$
  - (B) the message "failure"
- If  $a \geq \gamma$ , the algorithm outputs (A) with probability at least  $1 - \delta$ , making  $O\left(\frac{\log \delta^{-1}}{\sqrt{a}}\right)$  queries to  $A$ .
- If  $a < \gamma$ , the algorithm outputs either (A) or (B), making  $O\left(\frac{\log \delta^{-1}}{\sqrt{\gamma}}\right)$  queries to  $A$ .

The proof is given in Section 2.4 in [1]. The procedure of QAA( $A, \gamma, \delta$ ) is presented in Algorithm 1.

---

**Algorithm 1** QAA( $A, \gamma, \delta$ ), a modified version of **QSearch** in [26] with  $c = 3/2$

---

**Input:**  $A$  in Eq. (13),  $G$  in Eq. (14),  $\gamma \in (0, 1)$ ,  $\delta \in (0, 1)$

- 1: Set  $L = \left\lceil \log_{\frac{3}{2}} \frac{3}{4\sqrt{\gamma}} \right\rceil$ ,  $m_1 = \left\lceil \log_{\frac{3}{4}} \delta \right\rceil$  and  $m_2 = \left\lceil \log_{\frac{3}{8}} \delta \right\rceil$ .
- 2: **for**  $i = 1$  to  $m_1$  **do**
- 3:   Generate  $|\Phi\rangle$  and measure the single-qubit register. If the outcome is 1, output (A).
- 4: **end for**
- 5: **for**  $l = 1$  to  $L$  **do**
- 6:   Randomly choose an integer  $j \in [M]$ , where  $M = \left\lceil \left(\frac{3}{2}\right)^l \right\rceil$ .
- 7:   **for**  $i = 1$  to  $m_2$  **do**
- 8:     Generate  $G^j|\Phi\rangle$  and measure the single-qubit register. If the outcome is 1, output (A).
- 9:   **end for**
- 10: **end for**
- 11: Output (B).

---

Here,  $G$ , the so-called Grover operator, is defined as

$$G := -AS_0A^{-1}S_\chi. \quad (14)$$

$S_\chi$  is an operator that acts as  $S_\chi|\psi\rangle|0\rangle = |\psi\rangle|0\rangle$  and  $S_\chi|\psi\rangle|1\rangle = -|\psi\rangle|0\rangle$ , where  $|\psi\rangle$  is any state on the  $n$ -qubit register, and implemented just as a Z gate on the single-qubit register. The operator  $S_0$  acts as  $S_0|0\rangle|0\rangle = -|0\rangle|0\rangle$  and  $S_0|\Psi\rangle = |\Psi\rangle$  for any other computational basis states  $|\Psi\rangle$  on the system. We can implement this using a multi-controlled Z gate.

Let us see how Algorithm 1 works. We can show that, for any  $j \in \mathbb{N}$ ,

$$G^j|\Phi\rangle = \sin((2j+1)\theta_a)|\phi_1\rangle|1\rangle + \cos((2j+1)\theta_a)|\phi_0\rangle|0\rangle \quad (15)$$

holds, where  $\theta_a = \arcsin(\sqrt{a})$  [26]. Therefore, operating  $G$   $O(1/\sqrt{a})$  times on  $|\Phi\rangle$  makes the amplitude of  $|\phi_1\rangle|1\rangle$  of order 1, which means high probability to obtain 1 on the single-qubit register.

Let us make some more comments on QAA. First, note that QAA can be in fact regarded as an extension of Grover's algorithm, since the search problem in Theorem 1 can be solved by QAA. This is because we can generate the following state by  $O_N^{\text{EqPr}}$  and  $O_F$

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle|F(x)\rangle = \frac{1}{\sqrt{N}} |x_{\text{tar}}\rangle|1\rangle + \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n \setminus \{x_{\text{tar}}\}} |x\rangle|0\rangle, \quad (16)$$

which is in the form of Eq. (13) with  $|\phi_1\rangle = |x_{\text{tar}}\rangle$ . Second, QAA provides a quadratic speedup like Grover's algorithm. Instead of QAA, we can repeat generating  $|\Phi\rangle$  and measuring the qubit until we get the measurement outcome 1 and the state  $|\phi_1\rangle$ . This naive way yields  $O(a^{-1})$  repetitions in expectation. Therefore, QAA is quadratically faster than this. Third, note that, in Theorem 2, the marked state is defined as a state in which some qubit takes  $|1\rangle$ . Although in the original algorithm the marked state can be set more generally [26], the above setting is sufficient for the proposed algorithm for GW matched filtering, as we will see later.

**E. Quantum amplitude estimation**

Based on QAA, we can construct an algorithm called QAE for estimating the amplitude of a target state in a superposition state, or, more specifically,  $a$  in the state like Eq. (13) [26]. Roughly speaking, in the algorithm, we generate a superposition of states in the form of  $G^j |\Phi\rangle$  with various values of  $j$  by iteratively operating  $G$  controlled by some register  $R_{\text{QFT}}$ , and outputs an approximation of  $\theta_a$  onto  $R_{\text{QFT}}$  by quantum Fourier transform (QFT). Here, we do not enter details of the procedure but just present the following theorem, which is a modification of Theorem 12 in [26], without proof.

**Theorem 3.** *Suppose that we are given an access to an oracle  $A$  in Eq. (13). Then, for any integer  $t$  larger than 2, there is an oracle  $\tilde{O}_{A,t}^{\text{QAE}}$  that acts as  $\tilde{O}_{A,t}^{\text{QAE}} |0\rangle = \sum_{y \in \mathcal{Y}} \alpha_y |y\rangle$ , where some ancillary qubits are undisplayed. Here,  $\mathcal{Y}$  is a finite set of real numbers that includes a subset  $\tilde{\mathcal{Y}}$  consisting of elements  $\tilde{a}$  satisfying*

$$|\tilde{a} - a| \leq \frac{2\pi \sqrt{a(1-a)}}{t} + \frac{\pi^2}{t^2}, \quad (17)$$

and  $\{\alpha_y\}_{y \in \mathcal{Y}}$  are complex numbers satisfying  $\sum_{\tilde{y} \in \tilde{\mathcal{Y}}} |\alpha_{\tilde{y}}|^2 \geq 8/\pi^2$ . In  $\tilde{O}_{A,t}^{\text{QAE}}$ ,  $O_X$  is used  $O(t)$  times and  $O(n + \log t)$  qubits are used.

Here are some comments. In the original algorithm **Est\_Amp** in [26], we measure the state  $\sum_{y \in \mathcal{Y}} \alpha_y |y\rangle$  and obtain an estimate on  $a$ . However, we now stop the procedure at generation of the state, since, as explained below, we use QAE as a subroutine for the SNR calculation in searching high SNR templates by QAA and thus require it to be a unitary operation. Besides, note that the statement on qubit number is obvious since **Est\_Amp** in [26] uses only the register  $R_{\text{QFT}}$ , which has  $O(\log t)$  qubits, along with the system on which  $A$  acts.

We often want to enhance the lower bound  $8/\pi^2$  on the success probability of QAE to a given high value. We can accomplish this thanks to the following theorem, which is Lemma 1 in [15] and originally Lemma 6.1 in [27].

**Theorem 4.** *Let  $\mu \in \mathbb{R}$  and  $\epsilon \in \mathbb{R}_+$ . Let  $\mathcal{A}$  be an algorithm that outputs an  $\epsilon$ -approximation of  $\mu$  with probability  $\gamma \geq \frac{3}{4}$ . Then, for any  $\delta \in (0, 1)$ , the median of outputs in  $12 \lceil \log \delta^{-1} \rceil + 1$  runs of  $\mathcal{A}$  is an  $\epsilon$ -approximation of  $\mu$  with probability at least  $1 - \delta$ .*

This implies the following. Letting  $N$  be an integer larger than  $12 \lceil \log \delta^{-1} \rceil + 1$ , we generate the state

$$|\Psi_N\rangle := \sum_{y_1, \dots, y_N \in \mathcal{Y}} \alpha_{y_1} \cdots \alpha_{y_N} |y_1\rangle \cdots |y_N\rangle |\text{med}(y_1, \dots, y_N)\rangle \quad (18)$$

by operating  $\tilde{O}_{A,t}^{\text{QAE}}$  on each of the first  $N$  registers and using

$O_N^{\text{med}}$ . We rewrite this state as

$$\begin{aligned} |\psi_{\leq \epsilon}\rangle &:= \sum_{\substack{\tilde{a} \in \mathcal{Z} \\ |\tilde{a} - a| \leq \frac{2\pi \sqrt{a(1-a)}}{t} + \frac{\pi^2}{t^2}}} \beta_{\tilde{a}} |\phi_{\tilde{a}}\rangle |\tilde{a}\rangle, \\ |\psi_{> \epsilon}\rangle &:= \sum_{\substack{\tilde{a} \in \mathcal{Z} \\ |\tilde{a} - a| > \frac{2\pi \sqrt{a(1-a)}}{t} + \frac{\pi^2}{t^2}}} \beta_{\tilde{a}} |\phi_{\tilde{a}}\rangle |\tilde{a}\rangle, \end{aligned} \quad (19)$$

where  $\mathcal{Z}$  is some finite set of real numbers,  $\{\beta_{\tilde{a}}\}_{\tilde{a} \in \mathcal{Z}}$  are complex numbers, and  $\{|\phi_{\tilde{a}}\rangle\}_{\tilde{a} \in \mathcal{Z}}$  are states on the first  $N$  registers. Then,  $|\alpha_{\leq \epsilon}|^2$ , the squared amplitude of the state  $|\psi_{\leq \epsilon}\rangle$ , in which the number  $\tilde{a}$  on the last register satisfies Eq. (17), is larger than  $1 - \delta$ . This technique is used in the quantum algorithm for Monte Carlo integration, which is explained next.

**F. Quantum Monte Carlo integration**

On the basis of QAE, we can construct a quantum algorithm for Monte Carlo integration [15], which we hereafter call QMCI. Although Monte Carlo integration is generally a method to estimate integrals, we now consider it as a method to estimate the mean of  $X$  a given set of real numbers, since it is sufficient for the proposed algorithm. Among some versions presented in [15], we use the one for the situation where an upper bound on  $\text{Var}(X)$  is given.

**Theorem 5.** *Let  $N \in \mathbb{N}$  and  $X$  be a set of  $N$  real numbers,  $X_0, \dots, X_{N-1}$ , whose mean is  $\mu := \frac{1}{N} \sum_{i=0}^{N-1} X_i$  and sample variance  $\text{Var}(X)$  satisfies  $\text{Var}(X) \leq \sigma^2$  with  $\sigma \in \mathbb{R}_+$ . Suppose that we are given an oracle  $O_X$  that acts on a system with  $O(\log N)$  qubits in total as*

$$O_X |i\rangle |0\rangle = |i\rangle |X_i\rangle, \quad (20)$$

for any  $i \in [N]_0$ . Let  $\epsilon \in (0, 4\sigma)$  and  $\delta \in (0, 1)$ . Then, there is an oracle  $O_{X,\epsilon,\delta,\sigma}^{\text{mean}}$  such that

$$O_{X,\epsilon,\delta,\sigma}^{\text{mean}} |0\rangle = \sum_{y \in \mathcal{Y}} \alpha_y |y\rangle, \quad (21)$$

where some ancillary qubits are undisplayed. Here,  $\mathcal{Y}$  is a finite set of real numbers that includes a subset  $\tilde{\mathcal{Y}}$  consisting of  $\epsilon$ -approximations of  $\mu$  and  $\{\alpha_y\}_{y \in \mathcal{Y}}$  are complex numbers satisfying  $\sum_{\tilde{y} \in \tilde{\mathcal{Y}}} |\alpha_{\tilde{y}}|^2 \geq 1 - \delta$ . In  $O_{X,\epsilon,\delta,\sigma}^{\text{mean}}$ ,

$$O\left(\frac{\sigma}{\epsilon} \log^{3/2}\left(\frac{\sigma}{\epsilon}\right) \log \log\left(\frac{\sigma}{\epsilon}\right) \log\left(\frac{1}{\delta}\right)\right) \quad (22)$$

queries to  $O_X$  are made and

$$O\left(\left(\log N + \log\left(\frac{\sigma}{\epsilon}\right)\right) \log\left(\frac{\sigma}{\epsilon}\right) \log \log\left(\frac{\sigma}{\epsilon}\right) \log \delta^{-1}\right) \quad (23)$$

qubits are used.

The proof is presented in Appendix A in [1].

### III. QUANTUM ALGORITHM FOR GRAVITATIONAL WAVE MATCHED FILTERING

#### A. Problem and assumptions

Equipped with the above quantum algorithms, we now consider applying them to GW matched filtering. We start from formally stating the problem we consider.

**Problem 1.** Let  $T, M$  and  $N_{\text{temp}}$  be a positive real number, a positive even integer and a positive integer, respectively. Define  $\Delta t := T/M$  and, for  $k \in [M/2 - 1]$ ,  $f_k := k/T$ . Suppose that we are given a complex sequence  $\{\tilde{s}(f_k)\}_{k \in [M/2-1]}$ , a function  $S_n : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ , and, for every  $m \in [N_{\text{temp}}]_0$ , a function  $\tilde{Q}_m : \mathbb{R}_+ \rightarrow \mathbb{C}$ . Then, determine whether there exists any  $(m, j) \in [N_{\text{temp}}]_0 \times [M]_0$  such that  $\rho_{m,j}$  in Eq. (11) exceeds some given value or not. If there are such integer pairs, find one of them.

We need some preparations to tackle this problem. First, let us define some quantities for convenience. The first one is as follows: for  $\rho \in \mathbb{R}_+$ ,

$$\tilde{r}(\rho) := \frac{\#\{(m, j) \in [N_{\text{temp}}]_0 \times [M]_0 \mid \rho_{m,j} \geq \rho\}}{N_{\text{temp}}M}. \quad (24)$$

Obviously,  $\tilde{r}(\rho)$  represents the fraction of templates that give SNRs larger than  $\rho$  and is an analog of  $r(\rho)$  in Eq. (9). The next one is about the magnitude of template functions  $\tilde{Q}_m$  normalized by  $S_n$ :

$$\gamma := \max_{(m,k) \in [N_{\text{temp}}]_0 \times [\frac{M}{2}-1]} \frac{|\tilde{Q}_m(f_k)|}{\sqrt{S_n(f_k)\Delta t}}. \quad (25)$$

As discussed in Section 3.4.4 in [1], we expect that this is of order  $O(1)$ .

Next, let us make some assumptions needed to discuss the quantum algorithm and its complexity. The first one is about availability of some fundamental oracles.

**Assumption 1.** We have accesses to oracles  $O_{\text{Re}}$  and  $O_{\text{Im}}$  such that, for every  $(m, k) \in [N_{\text{temp}}]_0 \times [\frac{M}{2}]_0$ ,

$$\begin{aligned} O_{\text{Re}} |m\rangle |k\rangle |0\rangle &= \begin{cases} |m\rangle |k\rangle |0\rangle & ; \text{ if } k = 0 \\ |m\rangle |k\rangle \left| \Re \left( \frac{2\tilde{Q}_m^*(f_k)\tilde{s}(f_k)}{S_n(f_k)\Delta t} \right) \right\rangle & ; \text{ otherwise } \end{cases}, \\ O_{\text{Im}} |m\rangle |k\rangle |0\rangle &= \begin{cases} |m\rangle |k\rangle |0\rangle & ; \text{ if } k = 0 \\ |m\rangle |k\rangle \left| \Im \left( \frac{2\tilde{Q}_m^*(f_k)\tilde{s}(f_k)}{S_n(f_k)\Delta t} \right) \right\rangle & ; \text{ otherwise } \end{cases} \end{aligned} \quad (26)$$

We will discuss how to implement these in Section III D 2. The next assumption is on the mean and the variance of detector outputs used in the SNR calculation.

**Assumption 2.**

$$\begin{aligned} \left( \text{Mean} \left( \left\{ \frac{2\Re \tilde{s}(f_k)}{\sqrt{S_n(f_k)T}} \right\}_{k \in [\frac{M}{2}-1]} \right) \right)^2 &\leq 1, \\ \left( \text{Mean} \left( \left\{ \frac{2\Im \tilde{s}(f_k)}{\sqrt{S_n(f_k)T}} \right\}_{k \in [\frac{M}{2}-1]} \right) \right)^2 &\leq 1, \\ \text{Var} \left( \left\{ \frac{2\Re \tilde{s}(f_k)}{\sqrt{S_n(f_k)T}} \right\}_{k \in [\frac{M}{2}-1]} \right) &\leq 4, \text{Var} \left( \left\{ \frac{2\Im \tilde{s}(f_k)}{\sqrt{S_n(f_k)T}} \right\}_{k \in [\frac{M}{2}-1]} \right) &\leq 4. \end{aligned} \quad (27)$$

The validity of this assumption is discussed in Section 3.4.3 in [1]. As we will see in Section III C, this assumption is important for the proposed quantum algorithm, since it leads to the upper bound on the variance of summands in the SNR calculation, with which we can use QMCI for variables with bounded variance.

#### B. Previous algorithm

Before the new quantum algorithm, we review the algorithm proposed in [3]. It is shown as Algorithm 2, which is a modified version of Algorithm 2 in [3]. Given a SNR threshold  $\rho_{\text{th}}$ , this algorithm outputs the message "there is a signal" and  $m \in [N_{\text{temp}}]_0$  such that  $\rho_m \geq \rho_{\text{th}}$  with probability at least  $1 - \delta$ , if there exists such  $m$ . Note that Algorithm 2 in our paper uses QAA instead of QAE on  $O'_{\text{FFT}}|0\rangle|0\rangle|0\rangle$  in Algorithm 2 of [3], which does not affect the scaling of the complexity on  $M$  and  $N_{\text{temp}}$ . Also note that we can implement  $O_{\text{FFT}}$  in Eq. (28) by arithmetic oracles, since FFT is actually a sequence of arithmetic operations<sup>4</sup>.

The number of queries to arithmetic oracles in this algorithm is<sup>5</sup>

$$O\left(\frac{M \log M}{\sqrt{r(\rho_{\text{th}})}}\right), \quad (30)$$

when  $r(\rho_{\text{th}}) > 0$  and

$$O(\sqrt{N_{\text{temp}}}M \log M) \quad (31)$$

when  $r(\rho_{\text{th}}) = 0$ . We can see this as follows. QAA makes  $O(1/\sqrt{r(\rho_{\text{th}})})$  calls to  $O'_{\text{FFT}}$  and thus to  $O_{\text{FFT}}$  when  $r(\rho_{\text{th}}) > 0$ , and  $O(\sqrt{N_{\text{temp}}})$  calls to them when  $r(\rho_{\text{th}}) = 0$ . Besides, the number of queries to arithmetic oracles in  $O_{\text{FFT}}$  is of order  $O(M \log M)$  like the number of floating-point operations in FFT on a classical computer. Combining these, we get

<sup>4</sup> In fact, implementation of FFT as a quantum circuit has been studied in [28].

<sup>5</sup> Although Eqs. (30) and (31) do not have a factor  $\log N_{\text{temp}}$ , while Eq. (35) in [3] has, we omit this reasonably assuming that  $M \log M \gg \log N_{\text{temp}}$ .

---

**Algorithm 2** Previous algorithm for GW matched filtering (modified)

---

**Input:**

$\delta \in (0, 1)$ .  $\rho_{\text{th}} \in \mathbb{R}_+$  the SNR threshold. An oracle  $O_{\text{FFT}}$  such that, for every  $m \in [N_{\text{temp}}]_0$ ,

$$O_{\text{FFT}} |m\rangle |0\rangle = |m\rangle |\rho_m\rangle. \quad (28)$$

1: Combining  $O_{N_{\text{temp}}}^{\text{EqPr}}$ ,  $O_{\text{FFT}}$  and a comparer, construct an oracle  $O'_{\text{FFT}}$  that acts as

$$O'_{\text{FFT}} |0\rangle |0\rangle |0\rangle = \frac{1}{\sqrt{N_{\text{temp}}}} \sum_{m=0}^{N_{\text{temp}}-1} |m\rangle |\rho_m\rangle |\mathbb{1}_{\rho_m \geq \rho_{\text{th}}}\rangle. \quad (29)$$

- 2: Run QAA  $\left(O'_{\text{FFT}}, \frac{1}{N_{\text{temp}}}, \delta\right)$ .  
 3: **if** we get the message “failure” **then**  
 4:     Output the message “there is no signal”.  
 5: **else**  
 6:     Measure the first register in the quantum state output by QAA and let the outcome be  $m$ .  
 7:     Calculate  $\rho_m$  classically by FFT.  
 8:     **if**  $\rho_m \geq \rho_{\text{th}}$  **then**  
 9:         Output the message “there is a signal” and  $m$ .  
 10:     **else**  
 11:         Output the message “there is no signal”.  
 12:     **end if**  
 13: **end if**
- 

the complexity bounds in Eqs. (30) and (31). They show a quadratic speedup over the classical complexity of Eqs. (8) and (10) with respect to  $1/r(\rho_{\text{th}})$  and  $N_{\text{temp}}$ .

When it comes to qubit number, Algorithm 2 uses  $O(M)$  qubits, since FFT calculates  $\rho_{m,0}, \dots, \rho_{m,M-1}$  simultaneously and thus use  $O(M)$  registers to store intermediate and final calculation results.

### C. Proposed algorithm and its complexity

#### 1. Idea

The previous algorithm for GW matched filtering explained above uses FFT for the SNR calculation. On the other hand, from the formula Eq. (11) for SNR, we conceive the following idea: can we use QMCI for the SNR calculation? As we will see later, we can construct an oracle  $O_\rho$  to calculate the summand  $\tilde{\rho}_{m,j,k}$  in the SNR calculation in Eq. (11) making  $O(1)$  uses of  $O_{\text{Re}}$  and  $O_{\text{Im}}$ , and thus apply QMCI following Theorem 5. If we can set a bound  $\sigma^2$  on the sample variance of  $\{\tilde{\rho}_{m,j,k}\}_{k=1, \dots, \frac{M}{2}-1}$  and the accuracy  $\epsilon$  in the SNR calculation, the query complexity with respect to  $O_{\text{Re}}$  and  $O_{\text{Im}}$  is of order  $\tilde{O}(\sigma/\epsilon)$ .

This QMCI-based approach has a benefit on the qubit number reduction. As shown in Eq. (23), the number of qubits QMCI uses scales on  $M$  as  $O(\text{polylog}M)$ , since the SNR given as Eq. (11) is a mean of  $O(M)$  terms. This means large reduction compared to the number required by FFT, which is of order  $O(M)$ . This provides a large benefit, since quantum

computers will have a limitation on qubit capacity even in the future as mentioned in Introduction.

When it comes to the query complexity, the QMCI-based method is roughly same as the previous FFT-based algorithm. One might concern that the proposed method might worsen the scaling on  $M$  due to the expansion of the space searched by QAA. Unlike FFT, which simultaneously calculates  $\rho_{m,j}$  for all  $j \in [M]_0$ , QMCI is performed for each  $j$ . Thus, the search in the parameter space of  $m \in [N_{\text{temp}}]_0$  to find a large  $\rho_m$  in the previous algorithm is replaced with the search in a larger parameter space of  $(m, j) \in [N_{\text{temp}}]_0 \times [M]_0$  to find a large  $\rho_{m,j}$ . This means that the iteration number in QAA increases from  $\tilde{O}(1/\sqrt{r(\rho_{\text{th}})})$  to  $\tilde{O}(1/\sqrt{\tilde{r}(\rho_{\text{th}})})$ , which is  $\tilde{O}(\sigma\sqrt{M}/\epsilon\sqrt{r(\rho_{\text{th}})})$  at maximum since  $\tilde{r}(\rho_{\text{th}}) \geq r(\rho_{\text{th}})/M$ . Combining the aforementioned query complexity of the SNR calculation by QMCI, we bound the total number of queries to  $O_{\text{Re}}$  and  $O_{\text{Im}}$  as  $\tilde{O}(\sigma\sqrt{M}/\epsilon\sqrt{r(\rho_{\text{th}})})$ . Fortunately, since we can set  $\sigma = O(\sqrt{M})$  and  $\epsilon = O(1)$  as we will see below, the query complexity is resultingly  $\tilde{O}(M/\sqrt{r(\rho_{\text{th}})})$ , which is same as the number of queries to arithmetic oracles in the FFT-based method. Note that this is a reasonable comparison, since  $O_{\text{Re}}$  and  $O_{\text{Im}}$  are constructed by arithmetic oracles, with the aid of the quantum random access memory (QRAM) [29], as explained later.

We should also note that the nature of QMCI causes the following issue. The output of QMCI inevitably accompanies an error, and thus, even if the SNR of a given template calculated by QMCI exceeds the threshold  $\rho_{\text{th}}$ , its true SNR might be below  $\rho_{\text{th}}$ . We may think that we can evade such a misjudge by setting the accuracy  $\epsilon$  in QMCI extremely small, but it comes with large complexity. Therefore, we need to reasonably set the accuracy: following the nature of the problem under consideration, we should derive the error tolerance for SNR and set the QMCI accuracy matching it.

For this, we propose the following way. We consider GW matched filtering as a system that alarms us when the detector output seems to contain a signal. Besides, we consider the two levels of SNR threshold denoted by  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$ , which have the following meaning.

- If some templates have SNR  $\rho \geq \rho_{\text{hard}}$  for a given detector output, we want to be alarmed with certainty.
- We never want to be falsely alarmed when all templates have SNR  $\rho < \rho_{\text{soft}}$ .
- When no template has SNR  $\rho \geq \rho_{\text{hard}}$  but some have  $\rho \in [\rho_{\text{soft}}, \rho_{\text{hard}})$ , it is not needed but fine to be alarmed.

In this situation, we set the QMCI accuracy to  $(\rho_{\text{hard}} - \rho_{\text{soft}})/2$  and judge a template as matched if its SNR calculated by QMCI exceeds  $(\rho_{\text{hard}} + \rho_{\text{soft}})/2$  and mismatched otherwise. In this strategy, with high probability, a template with a true SNR  $\rho \geq \rho_{\text{hard}}$  is judged as matched and that with a true SNR  $\rho < \rho_{\text{soft}}$  is judged as mismatched. A template with a true SNR  $\rho_{\text{soft}} \leq \rho < \rho_{\text{hard}}$  is an intermediate case where the data may contain a signal near the threshold and can be judged as either matched or mismatched due to the QMCI error. We present

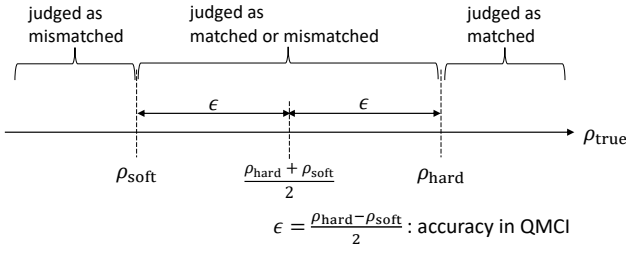


FIG. 1: An illustration of the current strategy. For every template, if the SNR calculated by QMCI exceeds  $(\rho_{\text{hard}} + \rho_{\text{soft}})/2$ , we judge it as matched, and otherwise we judge it as mismatched. With QMCI accuracy  $(\rho_{\text{hard}} - \rho_{\text{soft}})/2$ , this leads to a correct judgement for templates with a true SNR  $\rho_{\text{true}} < \rho_{\text{soft}}$  and  $\rho_{\text{true}} \geq \rho_{\text{hard}}$  with high probability.

an illustration of this strategy in Figure 1. We discuss how to set the two levels  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$  in Section III D 1.

Note that "match" and "mismatch" discussed here are the result of quantum computation for a given detector output, and do not necessarily mean the result correctly indicates whether there is a GW signal or not. In fact, a false detection can occur due to the random instrumental noise, which is not taken into account here. It is, on the other hand, relevant to how we set the thresholds of  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$ . See Section III D 1 for more detailed discussion.

Based on this idea, we design a quantum algorithm for GW matched filtering in the following part.

## 2. Main result

Then, the following is our main result, a new quantum algorithm for GW matched filtering and a theorem on its query complexity and the number of qubits used.

**Theorem 6.** *Under Assumptions 1 and 2, consider Problem 1. Let  $\rho_{\text{soft}}$  and  $\rho_{\text{hard}}$  be real numbers such that  $0 < \frac{\rho_{\text{hard}} - \rho_{\text{soft}}}{8\sqrt{5M}\gamma} < 4$ , and  $\delta$  be a real number in  $(0, 1)$ . Then, there is a quantum algorithm that uses*

$$O\left(\left(\log M + \log\left(\frac{\sqrt{M}\gamma}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right)\right)\log\left(\frac{\sqrt{M}\gamma}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right) \times \log\log\left(\frac{\sqrt{M}\gamma}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right)\log\left(\frac{N_{\text{temp}}M}{\delta}\right)\right) \quad (35)$$

qubits and behaves as follows:

- The algorithm outputs either of
  - (A) a message "there is a signal" and an integer pair  $(m, j) \in [N_{\text{temp}}]_0 \times [M]_0$  such that  $\rho_{m,j} \geq \rho_{\text{soft}}$ ,
  - (B) a message "there is no signal".
- If  $r(\rho_{\text{hard}}) > 0$ , the algorithm outputs (A) with probability at least  $1 - \delta$ . In the algorithm, the number of queries

### Algorithm 3 Proposed algorithm for GW matched filtering

- 1: Set  $\epsilon = \frac{\rho_{\text{hard}} - \rho_{\text{soft}}}{2}$ ,  $\sigma = 4\sqrt{5M}\gamma$  and  $\delta' = \frac{\delta}{4N_{\text{temp}}M}$ .
- 2: Combining  $O_{\text{Re}}$ ,  $O_{\text{Im}}$  and some arithmetic oracles, construct an oracles  $O_\rho$  such that, for every  $(m, j, k) \in [N_{\text{temp}}]_0 \times [M]_0 \times \left[\frac{M}{2}\right]_0$ ,

$$O_\rho |m\rangle |j\rangle |k\rangle |0\rangle = |m\rangle |j\rangle |k\rangle |\tilde{\rho}_{m,j,k}\rangle. \quad (32)$$

- 3: On the basis of Theorem 5, using  $O_\rho$ , construct an oracle  $O_{\rho, \epsilon, \delta', \sigma}^{\text{mean}}$  such that, for every  $(m, j) \in [N_{\text{temp}}]_0 \times [M]_0$ ,

$$O_{\rho, \epsilon, \delta', \sigma}^{\text{mean}} |m\rangle |j\rangle |0\rangle = |m\rangle |j\rangle \sum_{y \in \mathcal{Y}_{m,j}} \alpha_{\rho,y} |y\rangle, \quad (33)$$

where  $\mathcal{Y}_{m,j}$  is a finite set of real numbers that includes a subset  $\tilde{\mathcal{Y}}_{m,j}$  consisting of  $\epsilon$ -approximations of  $\rho_{m,j}$  and  $\{\alpha_y\}_{y \in \mathcal{Y}_{m,j}}$  are complex numbers satisfying  $\sum_{y \in \tilde{\mathcal{Y}}_{m,j}} |\alpha_{\rho,y}|^2 \geq 1 - \delta$ .

- 4: Construct an oracle  $O_{\text{AE}}$  that performs the following operation
 
$$\begin{aligned} &|0\rangle |0\rangle |0\rangle |0\rangle \\ &\rightarrow \sqrt{\frac{1}{N_{\text{temp}}M}} \sum_{m=0}^{N_{\text{temp}}-1} \sum_{j=0}^{M-1} |m\rangle |j\rangle |0\rangle |0\rangle \\ &\rightarrow \sqrt{\frac{1}{N_{\text{temp}}M}} \sum_{m=0}^{N_{\text{temp}}-1} \sum_{j=0}^{M-1} |m\rangle |j\rangle \sum_{y \in \mathcal{Y}_{m,j}} \alpha_{\rho,y} |y\rangle |0\rangle \\ &\rightarrow \sqrt{\frac{1}{N_{\text{temp}}M}} \sum_{m=0}^{N_{\text{temp}}-1} \sum_{j=0}^{M-1} |m\rangle |j\rangle \sum_{y \in \mathcal{Y}_{m,j}} \alpha_{\rho,y} |y\rangle (\mathbb{1}_{y \geq \rho_{\text{mid}}} |1\rangle + \mathbb{1}_{y < \rho_{\text{mid}}} |0\rangle), \end{aligned} \quad (34)$$

where  $\rho_{\text{mid}} := \frac{\rho_{\text{hard}} + \rho_{\text{soft}}}{2}$ . In Eq. (34), we use  $O_{N_{\text{temp}}}^{\text{EqPr}}$  and  $O_M^{\text{EqPr}}$  at the first arrow,  $O_{\rho, \epsilon, \delta', \sigma}^{\text{mean}}$  at the second arrow, and a comparer with  $|\rho_{\text{mid}}\rangle$  on an undisplayed ancillary register at the last arrow.

- 5: Run QAA  $\left(O_{\text{AE}}, \frac{1}{2N_{\text{temp}}M}, \frac{\delta}{2}\right)$ .
- 6: **if** we get the message "failure" **then**
- 7:     Output the message "there is no signal".
- 8: **else**
- 9:     Measure the first two registers in the quantum state output by QAA and let the outcome be  $(m, j)$ .
- 10:     Calculate  $\rho_{m,j}$  classically.
- 11:     **if**  $\rho_{m,j} \geq \rho_{\text{soft}}$  **then**
- 12:         Output the message "there is a signal" and  $(m, j)$ .
- 13:     **else**
- 14:         Output the message "there is no signal".
- 15:     **end if**
- 16: **end if**

to  $O_{\text{Re}}$  and  $O_{\text{Im}}$  is of order

$$O\left(\frac{\gamma\sqrt{M}}{(\rho_{\text{hard}} - \rho_{\text{soft}})\sqrt{r(\rho_{\text{hard}})}} \log^{3/2}\left(\frac{\gamma\sqrt{M}}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right) \times \log\log\left(\frac{\gamma\sqrt{M}}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right)\log\left(\frac{N_{\text{temp}}M}{\delta}\right)\log\delta^{-1}\right), \quad (36)$$

and thus

$$O\left(\frac{\gamma M}{(\rho_{\text{hard}} - \rho_{\text{soft}})\sqrt{r(\rho_{\text{hard}})}} \log^{3/2}\left(\frac{\gamma\sqrt{M}}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right) \times \log\log\left(\frac{\gamma\sqrt{M}}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right)\log\left(\frac{N_{\text{temp}}M}{\delta}\right)\log\delta^{-1}\right). \quad (37)$$



- If  $r(\rho_{\text{soft}}) = 0$ , the algorithm outputs (B) with certainty. In the algorithm, the number of queries to  $O_{\text{Re}}$  and  $O_{\text{Im}}$  is of order

$$O\left(\frac{\gamma M \sqrt{N_{\text{temp}}}}{\rho_{\text{hard}} - \rho_{\text{soft}}} \log^{3/2}\left(\frac{\gamma \sqrt{M}}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right) \log \log\left(\frac{\gamma \sqrt{M}}{\rho_{\text{hard}} - \rho_{\text{soft}}}\right) \times \log\left(\frac{N_{\text{temp}} M}{\delta}\right) \log \delta^{-1}\right). \quad (38)$$

- If  $r(\rho_{\text{hard}}) = 0$  and  $r(\rho_{\text{soft}}) > 0$ , the algorithm outputs either (A) or (B). In the algorithm, the number of queries to  $O_{\text{Re}}$  and  $O_{\text{Im}}$  is of order as in Eq. (38).

Here, we only present the algorithm as Algorithm 3. For the complete proof, especially that for the statements on the query complexity and the qubit number, see the proof in Appendix B in [1]. Since  $\text{Var}(\{\tilde{\rho}_{m,j,k}\}_{k=0,\dots,M/2-1}) \leq 80M\gamma^2$  holds for every  $(m, j) \in [N_{\text{temp}}]_0 \times [M]_0$  under Assumption 2 (see Lemma 1 in [1]), we can set  $\sigma$ , the upper bound on the variance of  $\{\tilde{\rho}_{m,j,k}\}_{k=0,\dots,M/2-1}$ , as in line 1.

#### D. Remarks on settings and assumptions

Now, we discuss the validity on the settings and assumptions in the algorithm proposed above.

##### 1. Setting of two thresholds

In Algorithm 3, we set the two SNR thresholds,  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$ , whose meanings are explained in Section III C 1. We consider the following is a plausible way to set them. First, let us denote a common value of the SNR threshold as  $\rho_{\text{com}}$ , e.g.  $\rho_{\text{com}} = 8^6$  [8]. Note that the calculated SNR, in general, has fluctuations due to the random detector noise, and its variance is 1 under the current normalization of the templates. Thus, even without the QMCI error, events with  $\rho \in [\rho_{\text{com}}, \rho_{\text{com}} + 1)$  could have a true SNR value smaller than the threshold, while events with  $\rho > \rho_{\text{com}} + 1$  are very likely to exceed the threshold. In light of this, it is reasonable to set  $\rho_{\text{soft}} = \rho_{\text{com}}$  and  $\rho_{\text{hard}} = \rho_{\text{com}} + 1$ , that is,  $\rho_{\text{soft}} = 8$  and  $\rho_{\text{hard}} = 9$  for  $\rho_{\text{com}} = 8$ . In this setting, Algorithm 3 detects events with  $\rho \geq 9$  with high probability and never falsely alarms us for events with  $\rho < 8$ , and events with  $\rho \in [8, 9)$  are detected or missed depending on fluctuations by the detector noise and the QMCI error.

##### 2. Implementation of $O_{\text{Re}}$ and $O_{\text{Im}}$

Here, we discuss the validity of Assumption 1, that is, implementability of  $O_{\text{Re}}$  and  $O_{\text{Im}}$ . If we have accesses to the

<sup>6</sup> For the detail of setting the SNR threshold, see Ref. [30] and Chap.7 of Ref. [31].

following oracles  $O_{\text{hSRe}}$ ,  $O_{\text{hSIm}}$ ,  $O_{\text{QRe}}$  and  $O_{\text{QIm}}$  such that, for every  $(m, k) \in [N_{\text{temp}}]_0 \times \left[\frac{M}{2} - 1\right]$ ,

$$O_{\text{hSRe}} |k\rangle |0\rangle = |k\rangle \left| \Re\left(\frac{\tilde{h}(f_k)}{S_n(f_k)}\right) \right\rangle, O_{\text{hSIm}} |k\rangle |0\rangle = |k\rangle \left| \Im\left(\frac{\tilde{h}(f_k)}{S_n(f_k)}\right) \right\rangle \quad (39)$$

and

$$O_{\text{QRe}} |m\rangle |k\rangle |0\rangle = |m\rangle |k\rangle \left| \Re \tilde{Q}_m(f_k) \right\rangle, O_{\text{QIm}} |m\rangle |k\rangle |0\rangle = |m\rangle |k\rangle \left| \Im \tilde{Q}_m(f_k) \right\rangle, \quad (40)$$

we can combine these along with arithmetic oracles to construct  $O_{\text{Re}}$  and  $O_{\text{Im}}$  (note that the remaining factor  $2/\Delta t$  is just a known real number independent of  $m$  and  $k$ ).

$O_{\text{QRe}}$  and  $O_{\text{QIm}}$  are in fact implementable. To see this, note that  $\tilde{Q}_m(f_k)$  is given as an explicit function of intrinsic parameters and  $f_k = k/T$  by theories of GW sources such as compact binary coalescences [4–7]. Therefore, if we can relate the index  $m$  to intrinsic parameter values by some elementary function, which is in fact possible under simple lattice-like template spacing such as [5], we can write  $\tilde{Q}_m$  as an explicit function of  $m$  and  $k$  and thus construct  $O_{\text{QRe}}$  and  $O_{\text{QIm}}$  using arithmetic oracles.

On the other hand,  $\tilde{h}(f_k)/S_n(f_k)$  is a factor determined by the experimental data and not represented by an explicit function. We therefore resort to QRAM [29]. This enables us to access  $N$  recorded data  $x_k$  labeled by  $k \in [N]_0$  and load a specified entry onto a register as

$$|k\rangle |0\rangle \rightarrow |k\rangle |x_k\rangle \quad (41)$$

in superposition in  $O(\log N)$  time. It takes  $O(M)$  time to register  $\left\{ \Re\left(\frac{\tilde{h}(f_k)}{S_n(f_k)}\right) \right\}_{k=1,\dots,\frac{M}{2}-1}$  and  $\left\{ \Im\left(\frac{\tilde{h}(f_k)}{S_n(f_k)}\right) \right\}_{k=1,\dots,\frac{M}{2}-1}$  into a QRAM in advance of running Algorithm 3, but this is expected to be less time-consuming than Algorithm 3 itself, which has  $\tilde{O}(\sqrt{N_{\text{temp}}}M)$  query complexity.

## IV. SUMMARY

In this paper, we proposed a new quantum algorithm for GW matched filtering based on Ref. [3] which has investigated the application of Grover’s search algorithm. Our method, described in details in Sec. III C, uses QMCI for the SNR calculation of Eq. (11) instead of FFT used in the classical method and in Ref. [3], and searches high SNR templates by QAA, running QMCI as a subroutine. To deal with the erroneous nature of QMCI, we propose to set two thresholds  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$  such that the proposed algorithm returns “there is a signal” for events with SNR  $\rho \geq \rho_{\text{hard}}$  with high probability and “there is no signal” for events with SNR  $\rho < \rho_{\text{soft}}$  with certainty.

Our main results are summarized in Eqs. (36), (37) and (38). By neglecting the logarithmic dependencies, we can see that the proposed algorithm has  $\tilde{O}(M/\sqrt{r(\rho_{\text{th}})})$  and  $\tilde{O}(M\sqrt{N_{\text{temp}}})$  query complexity for  $\tilde{r}(\rho_{\text{hard}}) > 0$  and  $\tilde{r}(\rho_{\text{hard}}) = 0$ , respectively. This still indicates a quadratic speedup with

respect to the template number  $N_{\text{temp}}$  and the same order of complexity with respect to the number of time-series data points  $M$  compared to the algorithm of Ref. [3], which is summarized in Sec. III B. We note that the choice of  $\rho_{\text{hard}}$  and  $\rho_{\text{soft}}$ , the accuracy of QMCI in other words, affects the complexity, but, according to the discussion in Sec. III D 1, we can take  $\rho_{\text{hard}} - \rho_{\text{soft}} = 1$ .

The advantage of this algorithm is that it requires only qubit number that logarithmically scales on  $M$  as described in Eq. (35), contrary to the FFT-based method that requires  $O(M)$  qubits. Therefore, this algorithm is expected to be beneficial in the situation that quantum computers have a limitation on qubit number, which is likely to occur due to the large overhead for quantum error correction. A possible drawback of the proposed algorithm compared to the algorithm in [3] is that, to load the detector output data onto a register in superposition in QMCI, it uses QRAM, whose experimental realization is challenging [32].

In any case, we believe that proposing multiple ways of applying quantum methods that have different pros and cons is highly meaningful in taking advantage of quantum computing in future GW experiments, given today's uncertainty on what the future quantum computers will be. Discussions for ap-

plying quantum computing in experimental physics have just started. We anticipate that more proposals will follow not only for GW data analysis but also for other heavy data analyses in various physical and astronomical experiments.

## ACKNOWLEDGEMENTS

G.M., S.K., and S.N. acknowledge support from the research project PGC2018-094773-B-C32, and the Spanish Research Agency (Agencia Estatal de Investigación) through the Grant IFT Centro de Excelencia Severo Ochoa No CEX2020-001007-S, funded by MCIN/AEI/10.13039/501100011033. S.K. is supported by the Spanish Atracción de Talento contract no. 2019-T1/TIC-13177 granted by Comunidad de Madrid, the I+D grant PID2020-118159GA-C42 of the Spanish Ministry of Science and Innovation and the i-LINK 2021 grant LINKA20416 of CSIC. T.Y. and S.K. are supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant no. JP20H01899 and JP20H05853. K.M. is supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant no. JPMXS0120319794 and JSPS KAKENHI Grant no. JP22K11924.

- 
- [1] K. Miyamoto et al. *arXiv preprint arXiv:2205.05966*, 2022.
  - [2] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2002.
  - [3] S. Gao et al. *Phys. Rev. Res.*, f4(2):023006, 2022.
  - [4] R. Balasubramanian, B. S. Sathyaprakash, and S. V. Dhurandhar. *Phys. Rev. D*, 53(6):3033, 1996.
  - [5] B. J. Owen. *Phys. Rev. D*, 53(12):6749, 1996.
  - [6] B. J. Owen and B. S. Sathyaprakash. *Phys. Rev. D*, 60(2):022002, 1999.
  - [7] B. Allen et al. *Phys. Rev. D*, 85(12):122006, 2012.
  - [8] B. P. Abbott et al. *Phys. Rev. Lett.*, 116(13):131103, 2016.
  - [9] B. P. Abbott et al. *Phys. Rev. X*, 9(3):031040, 2019.
  - [10] R. Abbott et al. *Phys. Rev. X*, 11(2):021053, 2021.
  - [11] R. Abbott et al. *arXiv preprint arXiv:2111.03606*, 2021.
  - [12] L. K. Grover. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
  - [13] J. W. Cooley, P. A. W. Lewis, and P. D. Welch. *Proceedings of the IEEE*, 55(10):1675, 1967.
  - [14] A. G. Fowler et al. *Phys. Rev. A*, 86(3):032324, 2012.
  - [15] A. Montanaro. *Proc. R. Soc. A*, 471(2181):20150301, 2015.
  - [16] C. Talbot and E. Thrane. *Phys. Rev. Research*, 2(4):043298, 2020.
  - [17] V. Vedral, A. Barenco, and A. Ekert. *Phys. Rev. A*, 54(1):147, 1996.
  - [18] E. Muñoz-Coreas and H. Thapliyal. *IEEE Transactions on Computers*, 68(5):729, 2018.
  - [19] H. Thapliyal et al. *IEEE Transactions on Emerging Topics in Computing*, 9(2):1045, 2019.
  - [20] T. Häner, M. Roetteler, and K. M. Svore. *arXiv preprint arXiv:1805.12445*, 2018.
  - [21] I. Koren. *Computer arithmetic algorithms*. AK Peters/CRC Press, 2001.
  - [22] D. J. Egger et al. *IEEE Transactions on Computers*, 70(12):2136, 2020.
  - [23] L. Grover and T. Rudolph. *arXiv preprint quant-ph/0208112*, 2002.
  - [24] W. H. Press et al. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.
  - [25] G. Brassard, P. Høyer, and A. Tapp. In *International Colloquium on Automata, Languages, and Programming*, pages 820–831. Springer, 1998.
  - [26] G. Brassard et al. *Contemporary Mathematics*, 305:53, 2002.
  - [27] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. *Theoretical Computer Science*, 43:169, 1986.
  - [28] R. Asaka, K. Sakai, and R. Yahagi. *Quantum Inf. Process.*, 19(8):277, 2020.
  - [29] V. Giovannetti, S. Lloyd, and L. Maccone. *Phys. Rev. Lett.*, 100(16):160501, 2008.
  - [30] C. Cutler et al. *Phys. Rev. Lett.*, 70:2984–2987, 1993.
  - [31] M. Maggiore. *Gravitational Waves. Vol. 1: Theory and Experiments*. Oxford Master Series in Physics. Oxford University Press, 2007.
  - [32] S. Arunachalam et al. *New J. Phys.*, 17(12):123010, 2015.