Quantum analog-digital conversion

Kosuke Mitarai, Masahiro Kitagawa, and Keisuke Fujii

Quantum analog-digital conversion

Kosuke Mitarai,[1, *] Masahiro Kitagawa,[1, 2] and Keisuke Fujii[3, 4, †]

[1] *Graduate School of Engineering Science, Osaka University,*
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan.
[2] *Quantum Information and Quantum Biology Division,*
Institute for Open and Transdisciplinary Research Initiatives,
Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan.
[3] *Graduate School of Science, Kyoto University, Yoshida-Ushinomiya-cho, Sakyo-ku, Kyoto 606-8302, Japan.*
[4] *JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan.*
(Dated: October 29, 2018)

Many quantum algorithms, such as Harrow-Hassidim-Lloyd (HHL) algorithm, depend on oracles that efficiently encode classical data into a quantum state. The encoding of the data can be categorized into two types; analog-encoding where the data are stored as amplitudes of a state, and digital-encoding where they are stored as qubit-strings. The former has been utilized to process classical data in an exponentially large space of a quantum system, whereas the latter is required to perform arithmetics on a quantum computer. Quantum algorithms like HHL achieve quantum speedups with a sophisticated use of these two encodings. In this work, we present algorithms that converts these two encodings to one another. While quantum digital-to-analog conversions have implicitly been used in existing quantum algorithms, we reformulate it and give a generalized protocol that works probabilistically. On the other hand, we propose a deterministic algorithm that performs a quantum analog-to-digital conversion. These algorithms can be utilized to realize high-level quantum algorithms such as a nonlinear transformation of amplitudes of a quantum state. As an example, we construct a "quantum amplitude perceptron", a quantum version of the neural network, and hence has a possible application in the area of quantum machine learning.

I. INTRODUCTION

A wide variety of quantum algorithms that potentially give quantum speedups over classical computers has been proposed. The problems that are efficiently solved by existing quantum algorithms can be divided into two types: ones where input data of a problem is relatively small in size but the problem itself is hard classically, and ones where the input of a problem is exponentially large, making it hard to handle on classical computers. The formers are solved by algorithms such as Shor's factoring [1], or quantum chemistry calculations [2]. On the other hand, there are algorithms that solve problems categorized as the latter. They achieve quantum speedups only if an oracle that encodes N classical data in $O(\text{poly}(\log N))$ time exists [3]. A famous example is Harrow-Hassidim-Lloyd (HHL) algorithm [4], which is an algorithm to apply a inverse A^{-1} of a matrix A to a $N = 2^n$-dimensional vector $\{c_j\}_{j=1}^N$. It requires us to construct a quantum state:

$$\sum_{j=1}^{N} c_j \ket{j}, \tag{1}$$

where $\{c_j\}_{j=1}^N$ normalized to satisfy $\sum_{j=1}^N |c_j|^2 = 1$. Data encoding in the format of Eq. (1) is crucial for all HHL-based algorithms [5–8], and others [9, 10]. We will call the state of Eq. (1) as an "analog-encoded" state,

since data are encoded into analog quantities, that is, complex amplitudes of a quantum state. Here we define an analog-encoding unitary transformation $U_A(\{c_j\})$ by:

$$U_A(\{c_j\}) \ket{0} = \sum_j c_j \ket{j}. \tag{2}$$

Another approach is to encode m-bits of binary data into qubit-strings. Let N and $\boldsymbol{d}_j = \{d_j^{(k)}\}_{k=1}^m$ ($d_j^{(k)} = 0, 1$, $j = 1, \cdots, N$) be the number of binary data provided and the data bitstrings. In this approach, data are encoded as follows:

$$\frac{1}{\sqrt{N}} \sum_{j=1}^{N} \ket{j} \ket{\boldsymbol{d}_j} = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \ket{j} \ket{d_j^{(m)} \cdots d_j^{(1)}}. \tag{3}$$

We will call this state as a "digital-encoded" state. For example, quantum algorithms for solving semidefinite programs (SDPs) [11, 12] depends on this encoding. Similarly to an analog-encoding unitary transformation, we define a digital-encoding unitary transformation by

$$U_D(\{\boldsymbol{d_j}\}) \ket{j} \ket{0} = \ket{j} \ket{\boldsymbol{d}_j}. \tag{4}$$

U_D is often called quantum random access memory (QRAM). Refs. [13, 14] have provided a protocol which employs qutrits to speedup a memory call to $O(\log^2 N)$ two-body interaction gates. Their method is promising compared to a conventional method that requires $O(N)$ operations. It might be worth noting that in the context of QRAM, it is usually assumed that we already have "memory cells" which stores data in the form accessible from a quantum computer. The $O(\log^2 N)$ operations do not include the construction of them.

---

* mitarai@qc.ee.es.osaka-u.ac.jp
† fujii.keisuke.2s@kyoto-u.ac.jp

Many quantum algorithms sophisticatedly use these two types of encodings. For example, in HHL algorithm [4], an analog-encoded state Eq. (1) is put through quantum phase estimation algorithm that digitally encode eigenvalues $\{\lambda_j\}$ of a matrix A, and then the inverse of them are multiplied to the amplitudes by controlled rotations (Fig. 1 (b)); in quantum metropolis sampling [15], energy eigenvalue $\{E_j\}$ of a Hamiltonian is first digitally encoded by the phase estimation, and then the encoded energies are transferred to amplitudes in the form of $e^{-\beta(E_j - E_k)}$, again using controlled rotations.

In this paper, we investigate the relation between these two different encoding methods. Specifically, we concentrate on conversions between these two encodings; can you go from "digital-encoding" to "analog-encoding" (quantum digital-to-analog conversion, QDAC), or the other way around (quantum analog-to-digital conversion, QADC)? DAC and ADC play important roles in classical information processing, since digitally-stored data are easier to handle than analog data which physical systems generate and are driven by. QDAC and QADC can be regarded as quantum analogs of them, and therefore there is a possibility that they stimulate the construction of more sophisticated quantum algorithms.

First, we formulate these problems. It is shown that QDAC can be implemented probabilistically, and QADC deterministically. A special case of QDAC, in fact, has implicitly been employed in existing algorithms such as HHL and quantum metropolis samplings. We unify those techniques and give a generalized procedure. QDAC and QADC algorithms provide an insight into what should be done in digital- or analog-encodings. Also, as an application, we show that a QADC-QDAC combined method can be utilized to perform almost arbitrary nonlinear transformations of amplitudes of a quantum state. This result can be utilized, for example, for a purpose of constructing quantum machine learning algorithms.

This paper is organized as follows. First in Sec. II, we summarize the algorithms we use as subroutines and define QADC and QDAC problem. In Sec.III, we review QDAC procedures that have implicitly been utilized in existing algorithms. We also present the generalized QDAC procedure. Then in Sec.IV, we present an algorithm to perform QADC. In Sec. V, we provide applications of QADC, and show that QADC combined with QDAC provides us a way to perform a nonlinear transformation of amplitudes of a quantum state.

II. PRELIMINARY

Here we summarize some useful results from existing works along with definitions of terms. Throughout this paper $N = 2^n$ denotes the number of data.

Fact 1 (analog encoding unitary [16, 17])
For given classical data $\{x_j\}_{j=1}^N \in \mathbb{R}^N$ such that $\sum_{j=1}^N x_j^2 = 1$, a binary-tree like classical data structure
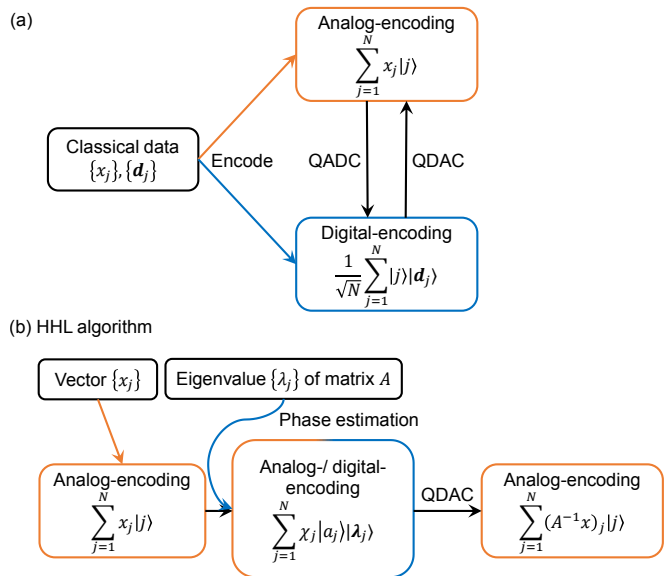

FIG. 1. (a) Schematic sketch of analog-encoding and digital-encoding. QDAC and QADC mediates these two encodings. (b) A brief flowchart of HHL algorithm [4]. $\{|a_j\rangle\}$ denote eigenvectors of a Hermitian matrix A, each corresponding to eigenvalues $\{\lambda_j\}$. χ_j are complex numbers such that $\sum_{j=1}^N x_j |j\rangle = \sum_{j=1}^N \chi_j |a_j\rangle$.

can be constructed in time $O(N \log^2 N)$ on a classical computer. With this structure, there exists a quantum algorithm that constructs an analog-encoding unitary $U_A(\{x_j\}_{j=1}^N)$ with $O(\log N \mathrm{poly}(\log \log N))$ single- and two-qubit gates.

We use the phase estimation algorithm stated below as a key ingredient of our QADC algorithm.

Fact 2 (phase estimation [18]) *Let U be a unitary operator acting on M-qubit Hilbert space with eigenstates $\{|\psi_j\rangle\}_{j=1}^{2^M}$ and corresponding eigenvalues $\{e^{2\pi i \phi_j}\}_{j=1}^{2^M}$ where $\phi_j \in [0, 1)$. Let $\epsilon = 2^{-m}$ for some positive integer m. There exists a quantum algorithm, which consists of $O(1/\epsilon)$ controlled-U calls and $O(\log^2(1/\epsilon))$ single- and two-qubit gates, that performs transformation $\sum_{j=1}^{2^M} a_j |\psi_j\rangle |0\rangle^{\otimes m} \to |\psi_{PE}\rangle = \sum_{j=1}^{2^M} a_j |\psi_j\rangle |\tilde{\phi}_j\rangle$ where $\tilde{\phi}_j$ denotes a bitstring $\tilde{\phi}_j^{(1)} \tilde{\phi}_j^{(2)} \cdots \tilde{\phi}_j^{(m)}$ such that $|\sum_{k=1}^m \tilde{\phi}_j^{(k)} 2^{-k} - \phi_j| \le \epsilon$ for all j with state fidelity at least $1 - \mathrm{poly}(\epsilon)$.*

We say that $|\tilde{\psi}\rangle$ has fidelity $1 - \delta$ with $|\psi\rangle$ when $|\langle \tilde{\psi} | \psi \rangle| = 1 - \delta$. The phase estimation algorithm can also be viewed as a digital-encoding unitary transformation, where the address is replaced by eigenstates of U.

Next we state a version of the amplitude amplification technique.

Fact 3 (amplitude amplification [19]) *Suppose we have a unitary operator U that acts on M-qubit Hilbert*

space as $U |0\rangle^{\otimes M} = \alpha |\psi\rangle |0\rangle + \beta |G\rangle |1\rangle$ where $|\psi\rangle, |G\rangle$ are arbitrary $M-1$-qubit states. Then probability of getting $|\psi\rangle |0\rangle$ can be made close to unity by $O(1/|\alpha|)$ application of U.

We define QDAC as follows.

Definition 1 (QDAC) *Let $\{d_j\}_{j=1}^N$ be a set of real numbers in $[0,1)$, each of which is represented by $d_j = \sum_{k=1}^m d_j^{(k)} 2^{-k}$ with binary variables $d_j^{(k)} \in \{0,1\}$. Let $\boldsymbol{d}_j$ denote the m-bit string $d_j^{(1)} \cdots d_j^{(m)}$. An m-bit QDAC operation transforms digital-encoded state $\frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\boldsymbol{d}_j\rangle$ to $C \sum_{j=1}^N d_j |j\rangle |0\rangle^{\otimes m}$ where C is a normalization constant.*

As for QADC, the amplitudes $\{c_j\}_{j=1}^N$ of a quantum state are, in general, complex numbers. Therefore we define three versions of QADC, each corresponding to the analog-to-digital conversion of the absolute value, the real part, and the imaginary part of $\{c_j\}$.

Definition 2 (abs-QADC) *Let $\tilde{\boldsymbol{r}}_j$ denote the m-bit string $\tilde{r}_j^{(1)} \cdots \tilde{r}_j^{(m)}$ that best approximates $|c_j|$ by $\sum_{k=1}^m \tilde{r}_j^{(k)} 2^{-k}$. An m-bit abs-QADC operation transforms analog-encoded state $\sum_{j=1}^N c_j |j\rangle |0\rangle^{\otimes m}$ to $\frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\tilde{\boldsymbol{r}}_j\rangle$.*

Definition 3 (real-QADC) *Let $\tilde{\boldsymbol{x}}_j$ denote the m-bit string $\tilde{x}_j^{(1)} \cdots \tilde{x}_j^{(m)}$ that best approximates the real part of c_j by $\sum_{k=1}^m \tilde{x}_j^{(k)} 2^{-k}$. An m-bit real-QADC operation transforms analog-encoded state $\sum_{j=1}^N c_j |j\rangle |0\rangle^{\otimes m}$ to $\frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\tilde{\boldsymbol{x}}_j\rangle$.*

Definition 4 (imag-QADC) *Let $\tilde{\boldsymbol{y}}_j$ denote the m-bit string $\tilde{y}_j^{(1)} \cdots \tilde{y}_j^{(m)}$ that best approximates the imaginary part of c_j by $\sum_{k=1}^m \tilde{y}_j^{(k)} 2^{-k}$. An m-bit imag-QADC operation transforms analog-encoded state $\sum_{j=1}^N c_j |j\rangle |0\rangle^{\otimes m}$ to $\frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\tilde{\boldsymbol{y}}_j\rangle$.*

We use quantum arithmetics as a subroutine, which is stated as the following theorem.

Fact 4 (quantum arithmetics [20]) *Let $\boldsymbol{a}, \boldsymbol{b}$ be m-bit strings. There exists a quantum algorithm that performs transformation $|\boldsymbol{a}\rangle |\boldsymbol{b}\rangle \to |\boldsymbol{a}\rangle |\boldsymbol{a} + \boldsymbol{b}\rangle$ or $|\boldsymbol{a}\rangle |\boldsymbol{b}\rangle \to |\boldsymbol{a}\rangle |\boldsymbol{ab}\rangle$ with $O(\mathrm{poly}(m))$ single- and two-qubit gates.*

Note that for accuracy defined as $\epsilon = 2^{-m}$, quantum arithmetics scales as $O(\mathrm{poly}(\log(1/\epsilon)))$. Furthermore, we assume the following statement as a fact.

Fact 5 *Some basic functions such as inverse, trigonometric functions, square root, and inverse trigonometric functions can be calculated to accuracy ϵ, that is, we can perform a transformation $|\boldsymbol{a}\rangle |0\rangle \to |\boldsymbol{a}\rangle |\tilde{\boldsymbol{f}}(\boldsymbol{a})\rangle$ such that $|\tilde{f}(a) - f(a)| \le \epsilon$ where $f(a)$ is the objective function, using $O(\mathrm{poly}(\log(1/\epsilon)))$ quantum arithmetics.*

A similar title is found on Ref. [21]. However, their purpose was to map a continuous-space wave function $|\psi\rangle = \int dx \psi(x) |x\rangle$ to a discrete-space wave function $|\psi_d\rangle = \sum_j \psi(j/N) |j\rangle$. In the context of this paper, it can be viewed as analog-to-analog conversion.

III. QDAC

It is actually straightforward to create an analog-encoded state from a digital-encoded state just by adding an ancilla qubit and performing a controlled rotation with the data register. In fact, this QDAC procedure has implicitly used in existing works. For example, HHL [4] has utilized the above protocol to multiply the inverse of eigenvalues of a Hermitian matrix A to an analog-encoded state vector. We state this QDAC operation formally as a theorem below.

Theorem 1 (QDAC with ancilla) *There exists a quantum algorithm that performs m-bit QDAC using $O(\mathrm{poly}(\log(1/\epsilon)))$ single- and two-qubit gates and one $U_D^\dagger$, where $\epsilon = 2^{-m}$, with probability $\sum_{j=1}^N d_j^2/N$.*

Proof - The procedure of the algorithm is as follows. Assume that we are provided with a digital-encoded state Eq. (3).

1. Compute $\varphi_j = \frac{2}{\pi} \cos^{-1} d_j$ by quantum arithmetics.

$$\frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\boldsymbol{d}_j\rangle |0\rangle^{\otimes m} \to \frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\boldsymbol{d}_j\rangle |\boldsymbol{\varphi}_j\rangle, \quad (5)$$

   where $\boldsymbol{\varphi}_j$ denotes m-bit strings $\varphi_j^{(1)} \cdots \varphi_j^{(m)}$ such that $\varphi_j = \sum_{k=1}^m \varphi_j^{(k)} 2^{-k}$.

2. Adding an ancilla qubit $|0\rangle_a$, perform a controlled rotation $R_y(\pi \varphi_j) = e^{i\pi \varphi_j Y/2}$, on the ancilla.

$$\frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\boldsymbol{d}_j\rangle |\boldsymbol{\varphi}_j\rangle |0\rangle_a$$

$$\to \frac{1}{\sqrt{N}} \sum_{j=1}^N |j\rangle |\boldsymbol{d}_j\rangle |\boldsymbol{\varphi}_j\rangle \left(d_j |0\rangle_a + \sqrt{1 - d_j^2} |1\rangle_a \right). \quad (6)$$

3. Measure the ancilla in computational basis. With probability $\sum_{j=1}^N d_j^2/N$, we obtain

$$C \sum_{j=1}^N d_j |j\rangle |\boldsymbol{d}_j\rangle |\boldsymbol{\varphi}_j\rangle |0\rangle_a, \quad (7)$$

   where $C = \sqrt{1/(\sum_{j=1}^N d_j^2)}$.

4. Uncompute φ_j (step 1) and apply $U_D^\dagger$. We now have an analog-encoded state

$$C \sum_{j=1}^N d_j |j\rangle. \quad (8)$$

Now we analyze the complexity. On step 1, $\cos^{-1} d_j$ can be calculated with quantum arithmetics using $O(\text{poly}(\log(1/\epsilon)))$ gates by Fact 5. Step 2 uses $m = \log(1/\epsilon)$ controlled rotations. Therefore, overall complexity for steps 1, 2 and 4 is $O(\text{poly}(\log(1/\epsilon)))$ and one $U_D^\dagger$. The success probability of step 3 is $\sum_{j=1}^N d_j^2/N$. $\square$

Success probability of above procedure can be rewritten in terms of the mean μ and the variance v of the data, since $\frac{1}{N}\sum_{j=1}^N d_j^2 = v + \mu^2$. Note that, when $v + \mu^2 \ll 1$ and the success probability is relatively small, the amplitude amplification technique can be utilized to shorten the expected running time quadratically from $O(1/(v+\mu^2))$ to $O\left(1/\sqrt{(v+\mu^2)}\right)$.

When one modifies Step 1 to give $\varphi_j = \cos^{-1}(f(d_j))$, we readily obtain the following.

Corollary 1.1 (generalized QDAC) *There exists a quantum algorithm that performs the transformation $\frac{1}{\sqrt{N}}\sum_{j=1}^N |j\rangle |\boldsymbol{d}_j\rangle \rightarrow C'\sum_{j=1}^N \tilde{f}(d_j)|j\rangle |0\rangle$ such that $|\tilde{f}(d_j) - f(d_j)| \le \epsilon$, where $f : [0,1) \to [0,1)$ is a function satisfying Fact 5, using $O(\text{poly}(\log(1/\epsilon)))$ single- and two-qubit gates, with probability $\sum_{j=1}^N \tilde{f}(d_j)^2/N = (NC')^{-1}$.*

If we choose $f(x) = \tanh(x)$, Collorary 1.1 provides an alternative way to implement a sigmoid function other than the one proposed in Ref. [22].

IV. QADC

First, we propose an abs-QADC algorithm. Note that the abs-QADC may easily be constructed with the real-QADC and the imag-QADC presented as Theorem 3 and 4. However, we expect that the abs-QADC algorithm that we present here would bring you some intuitions in the construction of the algorithm. We use the swap test [23], which is a special case of the Hadamard test, to extract the absolute value of amplitudes. The usual swap test, as described in Fig. 2, measures an absolute value of an inner product of arbitrary two states $|\psi\rangle$ and $|\xi\rangle$ as p_0, the probability of getting $|0\rangle$ from an ancilla qubit. If we input $|k\rangle$, which is a computational basis state, and an analog-encoded state to the swap test, we can extract a data x_k. The amplitude estimation [19] of p_0 can be utilized to encode the data digitally. An important trick used in the algorithm presented below is that this process can be parallelized.

Theorem 2 (abs-QADC) *There exists an m-bit abs-QADC algorithm that runs using $O(1/\epsilon)$ controlled-U_A gates and $O((\log^2 N)/\epsilon)$ single- and two-qubit gates with output state fidelity $1 - O(\text{poly}(\epsilon))$, where $\epsilon = 2^{-m}$.*

Proof - First we provide the algorithm. (See Fig. 3 for steps 1-4.)

1. Prepare address qubits. $\frac{1}{\sqrt{N}}\sum_{k=1}^N |k\rangle_{\text{ad}}$
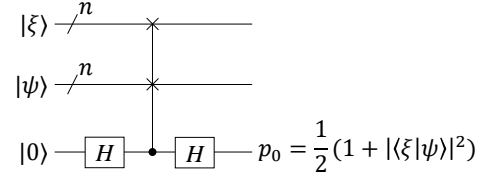

FIG. 2. Quantum circuit of the swap test [23].

2. Perform controlled-NOT from the address qubits to initialized ancilla qubits, which will be referred as qubits A, to get $\frac{1}{\sqrt{N}}\sum_{k=1}^N |k\rangle_{\text{ad}} |k\rangle_A$.

3. Prepare the analog-encoded state in data qubits, $\sum_j c_j |j\rangle_{\text{data}}$.

4. Using another ancilla qubit (We will call it qubit B), perform a swap test [23] without measurement between data-qubit and qubits A (Fig. 3). We have:

$$\sum_k \frac{|k\rangle_{\text{ad}}}{2\sqrt{N}}\left[\left(\sum_j c_j |j\rangle_{\text{data}}|k\rangle_A + |k\rangle_{\text{data}}\sum_j c_j |j\rangle_A\right)|0\rangle_B + \right.$$
$$\left.\left(\sum_j c_j |j\rangle_{\text{data}}|k\rangle_A - |k\rangle_{\text{data}}\sum_j c_j |j\rangle_A\right)|1\rangle_B\right] \quad (9)$$
$$\equiv \frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}|\Psi_k\rangle_{\text{data},A,B}. \quad (10)$$

Fig. 3 shows the quantum circuit from step 1 to step 4. We define V to be the combined unitary transformation of step 3 and 4. This step extracts an absolute value r_k of amplitude c_k each corresponding to an address $|k\rangle_{\text{ad}}$. The similar idea is also used in Ref. [24].

5. Construct a gate

$$G = V(\text{CNOT})_{\text{ad}\to A}S_0(\text{CNOT})_{\text{ad}\to A}V^\dagger Z_B, \quad (11)$$

where S_0 is a conditional phase shift gate; $S_0 = I - 2(|0\rangle\langle 0|)_{\text{data},A,B}$ and Z_B is a Pauli Z gate only acting on the qubit B (Fig. 4). The act of G can be written as

$$G\frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}|\Psi_k\rangle_{\text{data},A,B} = $$
$$\frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}\left(G_k |\Psi_k\rangle_{\text{data},A,B}\right), \quad (12)$$

where

$$G_k = V S_k V^\dagger Z_B, \quad (13)$$

and

$$S_k = I - 2(|0\rangle\langle 0|)_{\text{data},B} \otimes (|k\rangle\langle k|)_A. \quad (14)$$

Each $|\Psi_k\rangle_{\text{data},A,B}$ is decomposed into two of eigenstates $|\Psi_{k+}\rangle_{\text{data},A,B}$ and $|\Psi_{k-}\rangle_{\text{data},A,B}$ of G_k, each respectively corresponding to eigenvalue $\lambda_{k\pm} = e^{\pm i2\pi\theta_k}$ where $\sin(\pi\theta_k) = \sqrt{\frac{1}{2}(1+r_k^2)}$ and $\theta_k \in [1/4, 1/2]$. The decomposition is: $|\Psi_k\rangle_{\text{data},A,B} = \frac{-i}{\sqrt{2}}(e^{i\pi\theta_k}|\Psi_{k+}\rangle_{\text{data},A,B} - e^{-i\pi\theta_k}|\Psi_{k-}\rangle_{\text{data},A,B})$. See Appendix for detailed description.

6. Introducing the register qubits, run the phase estimation of G as depicted in Fig.5. Then we have:

$$\frac{1}{\sqrt{2N}}\sum_k |k\rangle_{\text{ad}}\Big(|\boldsymbol{\theta}_k\rangle_{\text{reg}'}|\Psi_{k+}\rangle_{\text{data},A,B}$$

$$+ |\mathbf{1}-\boldsymbol{\theta}_k\rangle_{\text{reg}'}|\Psi_{k-}\rangle_{\text{data},A,B}\Big)$$

$$\equiv \frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}|\Psi_{k,AE}\rangle_{\text{reg}',\text{data},A,B} \qquad (15)$$

where $|\boldsymbol{\theta}_k\rangle_{\text{reg}'}$ and $|\mathbf{1}-\boldsymbol{\theta}_k\rangle_{\text{reg}'}$ are m-bit strings that store θ_k and $1-\theta_k$ as binary data, and

$$|\Psi_{k,AE}\rangle_{\text{reg}',\text{data},A,B} =$$
$$\frac{1}{\sqrt{2}}\Big(|\boldsymbol{\theta}_k\rangle_{\text{reg}'}|\Psi_{k+}\rangle_{\text{data},A,B} + |\mathbf{1}-\boldsymbol{\theta}_k\rangle_{\text{reg}'}|\Psi_{k-}\rangle_{\text{data},A,B}\Big) \qquad (16)$$

7. On another register, using digital quantum arithmetics, calculate $r_k = \sqrt{2\sin^2\pi\theta_k - 1}$. Note that $\sin\pi\theta_k = \sin\pi(1-\theta_k)$, and r_k is uniquely recovered since $r_k \in [0,1]$. Then finally we get:

$$\frac{1}{\sqrt{N}}\sum_{k=0}^N |k\rangle_{\text{ad}}|\tilde{\boldsymbol{r}}_k\rangle_{\text{reg}}|\Psi_{k,AE}\rangle_{\text{reg}',\text{data},A,B}. \qquad (17)$$

8. Uncompute the data, A, B, reg' qubits. We obtain:

$$\frac{1}{\sqrt{N}}\sum_{k=0}^N |k\rangle_{\text{ad}}|\tilde{\boldsymbol{r}}_k\rangle_{\text{reg}}|0\rangle_{\text{reg}',\text{data},A,B}, \qquad (18)$$

which is a digital-encoded state.

Here we analyze the complexity of the above algorithm. For steps 1 to 4, we used $O(\log N)$ single- and two-qubit gate. On step 5, the phase estimation, we need to use $O(1/\epsilon)$ of controlled-U_A and $O(\log^2 N/\epsilon)$ of single- and two-qubit gate. Step 6, quantum arithmetics takes $O(\text{poly}(\log(1/\epsilon)))$ by Fact 5. Therefore, overall complexity is $O(1/\epsilon)$ of controlled-U_A and $O(\log^2 N/\epsilon)$ of single- and two-qubit gate. The fidelity of the output state is $1 - O(\text{poly}(\epsilon))$ by Fact 2. $\qquad \square$

Next we show the real-QADC.

Theorem 3 (real-QADC) *There exists an m-bit real-QADC algorithm that runs using $O(1/\epsilon)$ controlled-U_A gates and $O((\log^2 N)/\epsilon)$ single- and two-qubit gates with output state fidelity $1 - O(\text{poly}(\epsilon))$, where $\epsilon = 2^{-m}$.*
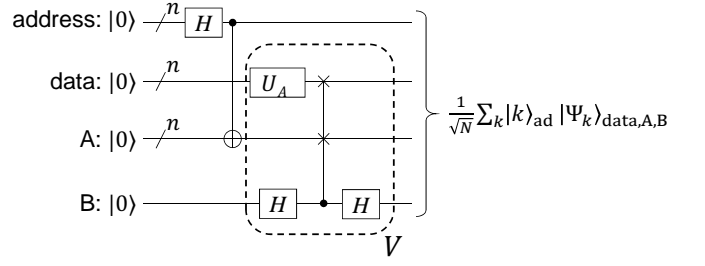

FIG. 3. Quantum circuit through steps 1 to 4 of abs-QADC in the main text.
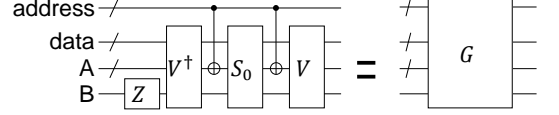

FIG. 4. Definition of gate G in abs-QADC.

Proof - We provide the algorithm. (See Fig. 3 for steps 1-3.) The algorithm presented here is slightly modified one from the previous algorithm for QADC.

1. Prepare address qubits. $\frac{1}{\sqrt{N}}\sum_{k=1}^N |k\rangle_{\text{ad}}$

2. Prepare the analog-encoded state in data qubits, $\sum_j x_j |j\rangle_{\text{data}}$

3. Using another ancilla qubit (We will call it qubit B), perform an Hadamard test as described in Fig. 6. We have:

$$\sum_k \frac{|k\rangle_{\text{ad}}}{2\sqrt{N}}\Bigg[\Bigg(\sum_j x_j |j\rangle_{\text{data}} + |k\rangle\Bigg)|0\rangle_B +$$

$$\Bigg(\sum_j x_j |j\rangle_{\text{data}} - |k\rangle\Bigg)|1\rangle_B\Bigg] \qquad (19)$$

$$\equiv \frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}|\Psi_k\rangle_{\text{data},B}. \qquad (20)$$

This step extracts the real part x_k of a complex amplitude c_k each corresponding to an address $|k\rangle_{\text{ad}}$. Fig. 6 shows the quantum circuit from step 1 to 3. We define W to be the combined unitary transformation of step 2 and 3.

4. Construct a gate

$$G' = WS_0'W^\dagger Z_B, \qquad (21)$$

where S_0' is a conditional phase shift gate; $S_0' = I - 2(|0\rangle\langle 0|)_{\text{data},B}$ and Z_B is a Pauli Z gate only acting on the qubit B (Fig. 4). The act of G' can be written as

$$G'\frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}|\Psi_k\rangle_{\text{data},B} =$$
$$\frac{1}{\sqrt{N}}\sum_k |k\rangle_{\text{ad}}\Big(G_k'|\Psi_k\rangle_{\text{data},B}\Big), \qquad (22)$$
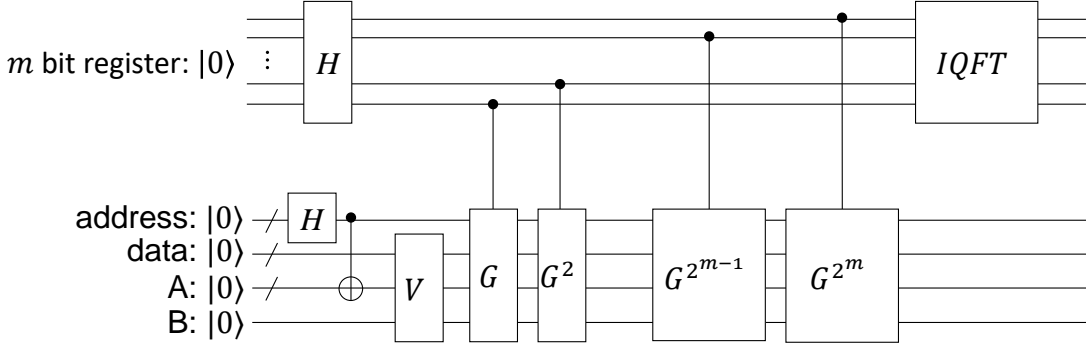
FIG. 5. Step 6 of the abs-QADC algorithm. The phase estimationa is performed to encode the analog-encoded value x_j into qubit bitstrings. IQFT is inverse quantum Fourier transformation [25].

where

$$G'_k = (1 - 2 |\Psi_k\rangle_{\text{data},B} \langle \Psi_k|_{\text{data},B}) Z_B. \qquad (23)$$

Each $|\Psi_k\rangle_{\text{data},B}$ is decomposed into two of eigenstates $|\Psi_{k+}\rangle_{\text{data},B}$ and $|\Psi_{k-}\rangle_{\text{data},B}$ of G_k, each respectively corresponding to eigenvalue $\lambda_{k\pm} = e^{\pm i 2\pi\theta_k}$ where $\sin(\pi\theta_k) = \sqrt{\frac{1}{2}(1 + x_k)}$ and $\theta_k \in [1/4, 1/2]$. The decomposition is: $|\Psi_k\rangle_{\text{data},B} = \frac{-i}{\sqrt{2}}(e^{i\pi\theta_k} |\Psi_{k+}\rangle_{\text{data},B} - e^{-i\pi\theta_k} |\Psi_{k-}\rangle_{\text{data},B})$. The detail of this transformation is similar to the one described in Appendix and thus omitted.

5. Introducing the register qubits, run the phase estimation of G'. Then we have:

$$\frac{1}{\sqrt{2N}} \sum_k |k\rangle_{\text{ad}} \left(|\boldsymbol{\theta}_k\rangle_{\text{reg}'} |\Psi_{k+}\rangle_{\text{data},B} \right.$$

$$\left. + |\mathbf{1} - \boldsymbol{\theta}_k\rangle_{\text{reg}'} |\Psi_{k-}\rangle_{\text{data},B} \right)$$

$$\equiv \frac{1}{\sqrt{N}} \sum_k |k\rangle_{\text{ad}} |\Psi_{k,AE}\rangle_{\text{reg}',\text{data},B} \qquad (24)$$

where $|\boldsymbol{\theta}_k\rangle_{\text{reg}'}$ and $|\mathbf{1} - \boldsymbol{\theta}_k\rangle_{\text{reg}'}$ are m-bit strings that store θ_k and $1 - \theta_k$ as binary data, and

$$|\Psi_{k,AE}\rangle_{\text{reg}',\text{data},B} =$$

$$\frac{1}{\sqrt{2}} \left(|\boldsymbol{\theta}_k\rangle_{\text{reg}'} |\Psi_{k+}\rangle_{\text{data},B} + |\mathbf{1} - \boldsymbol{\theta}_k\rangle_{\text{reg}'} |\Psi_{k-}\rangle_{\text{data},B} \right)$$

$$(25)$$

6. On another register, using digital quantum arithmetics, calculate $x_k = 2\sin^2 \pi\theta_k - 1$. Note that $\sin \pi\theta_k = \sin \pi(1 - \theta_k)$. Then finally we get:

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N} |k\rangle_{\text{ad}} |\tilde{\boldsymbol{x}}_k\rangle_{\text{reg}} |\Psi_{k,AE}\rangle_{\text{reg}',\text{data},B} . \qquad (26)$$

7. Uncompute the data, A, B, reg' qubits. We obtain:

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N} |k\rangle_{\text{ad}} |\tilde{\boldsymbol{x}}_k\rangle_{\text{reg}} |0\rangle_{\text{reg}',\text{data},B} , \qquad (27)$$
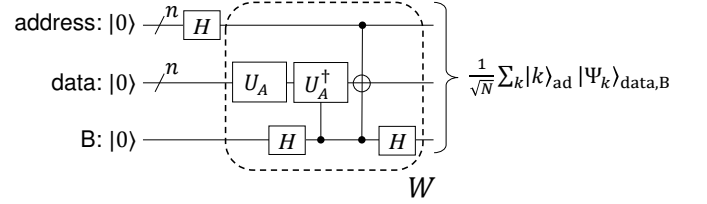

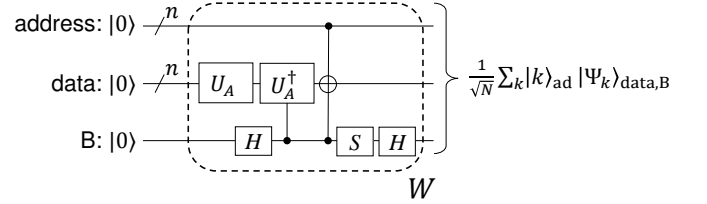FIG. 6. A quantum circuit through steps 1 to 3 of real-QADC in the main text.

FIG. 7. A quantum circuit element for imag-QADC. The imag-QADC can be performed by replacing W in the real-QADC by this circuit.

which is a digital-encoded state.

The runtime of this algorithm is as same as the abs-QADC. $\qquad \square$

The imag-QADC can be constructed in the same manner as the real-QADC. In this case, we replace the gate W to the one described in Fig. 7. Therefore we have the following.

Theorem 4 (imag-QADC) *There exists an m-bit imag-QADC algorithm that runs using $O(1/\epsilon)$ controlled-U_A gates and $O((\log^2 N)/\epsilon)$ single- and two-qubit gates with output state fidelity $1 - O(\text{poly}(\epsilon))$, where $\epsilon = 2^{-m}$.*

V. APPLICATIONS

A. Classical data loading

As stated in the introduction, there are some algorithms, such as quantum SDP solvers [11, 12], that require an oracle which encodes N classical data into a quantum state Eq. (3) in time $O(\text{poly}(\log N))$.

QADC can be utilized for this purpose. Assume that a binary-tree structure required in Fact 1 is already constructed on the classical side. Combined with the analog-encoding unitary U_A of Fact 1, the m-bit digital-encoded state $\frac{1}{\sqrt{N}} \sum_{j=1}^{N} |j\rangle |\boldsymbol{d}_j\rangle$ can be prepared with $O(\log^2 N \text{poly}(\log \log N)/\epsilon)$ quantum gates.

Recently another method for digital encoding has been proposed [26]. They proposed a protocol that directly encodes binary classical data into qubits.

B. Nonlinear transformation of amplitude

The transformation performed on the probability amplitudes c_j of a quantum state $\sum_{j=1}^{N} c_j |j\rangle$, without measurements, is always linear owing to the unitarity of the quantum dynamics. Even with (projective) measurement, the transformation is restricted to the form of c_j/C, where C is some constant that is determined by the normalization condition. Therefore, a transformation of the form $f(c_j)$ with an arbitrary function f cannot be done, without encoding them in digital form using QADC. We state the fact that this form of nonlinear transformation of amplitude can be performed using QADC and QDAC as the following theorem.

Theorem 5 *Let data $\{c_j\}_{j=1}^{N}$, analog-encoding unitary $U_A(\{c_j\}_{j=1}^{N})$ and analog-encoded state $\sum_{j=1}^{N} c_j |j\rangle$ be given. For any function $f : \mathbb{C} \rightarrow [-1,1]$ that satisfies Fact 5, there exists a probabilistic quantum algorithm that performs transformation $\sum_{j=1}^{N} c_j |j\rangle \rightarrow C' \sum_{j=1}^{N} \tilde{f}(c_j) |j\rangle$, such that $|\tilde{f}(c_j) - f(c_j)| \leq \epsilon$, with $O(1/\epsilon)$ controlled-U_A gates and $O((\log^2 N)/\epsilon)$ single- and two-qubit gates. Probability of success is $\sum_{j=1}^{N} \tilde{f}(c_j)^2/N$ and the output state fidelity is $1 - O(\text{poly}(\epsilon))$*

Proof - By running the QADC algorithm from Theorem 3 and Theorem 4 in parallel, we get a state $\frac{1}{\sqrt{N}} \sum_{j=1}^{N} |j\rangle |\tilde{\boldsymbol{x}}_j\rangle |\tilde{\boldsymbol{y}}_j\rangle$ using $O(1/\epsilon)$ controlled-U_A gates and $O((\log^2 N)/\epsilon)$ single- and two-qubit gates with state fidelity $1 - O(\text{poly}(\epsilon))$. Then performing the modified QDAC from Corollary 1.1, we get $C' \sum_{j=1}^{N} \tilde{f}(x_j, y_j) |j\rangle |\boldsymbol{x}_j\rangle |\boldsymbol{y}_j\rangle$ with $O(\text{poly}(\log(1/\epsilon)))$ single- and two-qubit gates and with probability $\sum_{j=1}^{N} \tilde{f}(c_j)^2/N$. Finally, application of inverse QADC leaves us the desired state. $\square$

The most useful application of this would be to deal with a "quantum big-data". For example, using U_A on two registers, we can prepare

$$U_A |0\rangle \otimes U_A |0\rangle = \sum_{i,j} c_i c_j |i\rangle |j\rangle . \qquad (28)$$

U_A is not restricted to the loading of classical data $\{c_j\}$, but can also be a time evolution operator e^{-iHt}, where H is some Hamiltonian which can efficiently be simulated on a quantum computer. Notice that the tensor product structure of quantum mechanics has calculated the product c_i, c_j and the combined system has N^2 of amplitudes. This nonlinearity is employed in quantum circuit learning [27] for machine learning application. The direct digital encoding of these N^2 dimensional data on a quantum state might be impractical, since the calculation of the product $\{c_i c_j\}_{i,j=1}^{N}$ takes $O(N^2)$ time and space classically. In comparison, the construction of the quantum state Eq. (28) requires only $O(N \log N)$ classical and $O(\log N \text{poly}(\log \log N))$ quantum operations. Tensor product structure is useful to introduce nonlinearity to the data in the form of Eq. (28), however, it cannot be used to introduce general and more complex transformation stated in above theorem.

Applying QADC on this combined state yields a state,

$$\sum_{i,j} |i\rangle |j\rangle |c_i c_j\rangle . \qquad (29)$$

Note that this transformation can also be performed in time $O(poly(\log N))$. Then applying the nonlinear QDAC procedure and the inverse of QADC, one gets,

$$\sum_{i,j} f(c_i c_j) |i\rangle |j\rangle |0\rangle , \qquad (30)$$

which has nonlinearly transformed probability amplitudes. A further extension is discussed in the next subsection.

C. Quantum amplitude perceptron

When one chooses $f(x) = \tanh(x)$ or the ReLU function which are frequently used in neural networks, it can readily mimic the perceptron, which is a building block of neural networks. We denote such an activation function by $\sigma(x)$ In neural network, a transformation of the form $\sigma(\boldsymbol{w} \cdot \boldsymbol{x})$, where $\boldsymbol{w}$ and $\boldsymbol{x}$ are a weight vector and a data vector, is utilized to learn some task. The training of the network is done by tuning the weight $\boldsymbol{w}$ to give some specific output. To mimic this, first we apply a parametrized unitary transformation $U(\boldsymbol{\theta})$, which corresponds to the weight $\boldsymbol{w}$, on an analog-encoded state $\sum_{i=1}^{N} c_i |i\rangle$, yielding a state $\sum_{i=1}^{N} \sum_{j=1}^{N} u_{jk}(\boldsymbol{\theta}) c_j |i\rangle$. Note that the amplitude x_i can be a resultant amplitude after the tensor product multiplication described in the previous section. Then use the procedure of Theorem 5 with

the activation function σ, which produces,

$$C' \sum_i \sigma \left[\sum_j u_{ij}(\boldsymbol{\theta}) x_j \right] |i\rangle. \tag{31}$$

The full tomography of this state would require us an exponential time, but if one is interested in the constant number of perceptron outputs, performing a swap test [23] with chosen basis $\{|k\rangle\}$ is enough to extract them. The state Eq. (31) can be further transformed by some unitary gates to determine the readout weight of the output.

For machine learning, it is necessary to optimize the parameter $\boldsymbol{\theta}$. Although the gradient of the output with respect to $\boldsymbol{\theta}$ cannot be extracted due to the complex form of Eq. (31), gradient-free methods can be utilized, just as same as mentioned in Ref. [22].

Finally, we note here, if one can determine the value of $\sum_i \left\{ \sigma \left[\left(\sum_j u_{ij}(\boldsymbol{\theta}) x_j \right)^2 \right] \right\}^2$ somehow, amplitude amplification technique can be employed to make QDAC procedure deterministic. This would enable us to implement a multilayer neural network on a quantum computer deterministically.

VI. CONCLUSION

We have formulated QDAC and QADC, and described algorithms to implement them on quantum computers. Although QDAC protocols have implicitly been utilized in existing algorithms, we have generalized it to facilitate complex nonlinear functions. We have also presented an algorithm that performs QADC. It was shown that a combination of QADC and QDAC enables us almost arbitrary nonlinear transformations of amplitudes of a quantum state. The possible application of this nonlinear transformation is to make a quantum amplitude perceptron, which can be employed to construct more sophisticated quantum machine learning algorithms.

Appendix: Eigenvalues and eigenstates of G_k

Here we calculate the eigenvalues and eigenstates of G_k defined in Eq. (22). We consider G_k acting on the state

$$|\Psi_k\rangle_{\text{data},A,B} = V |0\rangle_{\text{data}} |k\rangle_A |0\rangle_B =$$

$$\frac{1}{2} \left[\left(\sum_j c_j |j\rangle_{\text{data}} |k\rangle_A + |k\rangle_{\text{data}} \sum_j c_j |j\rangle_A \right) |0\rangle_B + \right.$$

$$\left. \left(\sum_j c_j |j\rangle_{\text{data}} |k\rangle_A - |k\rangle_{\text{data}} \sum_j c_j |j\rangle_A \right) |1\rangle_B \right]. \tag{A.1}$$

First, we define two normalized states

$$|\Psi_{k0}\rangle = \frac{1}{2\alpha_k} \left(\sum_j c_j |j\rangle_{\text{data}} |k\rangle_A + |k\rangle_{\text{data}} \sum_j c_j |j\rangle_A \right) |0\rangle_B, \tag{A.2}$$

$$|\Psi_{k1}\rangle = \frac{1}{2\beta_k} \left(\sum_j c_j |j\rangle_{\text{data}} |k\rangle_A - |k\rangle_{\text{data}} \sum_j c_j |j\rangle_A \right) |1\rangle_B, \tag{A.3}$$

where

$$\alpha_k = \sqrt{\frac{1}{2}(1 + r_k^2)}, \tag{A.4}$$

$$\beta_k = \sqrt{\frac{1}{2}(1 - r_k^2)}. \tag{A.5}$$

We define $\theta_k \in [1/4, 1/2)$ by

$$\sin \pi \theta_k = \alpha_k. \tag{A.6}$$

Then $|\Psi_k\rangle_{\text{data},A,B}$ can be rewritten as

$$|\Psi_k\rangle_{\text{data},A,B} = \alpha_k |\Psi_{k0}\rangle + \beta_k |\Psi_{k1}\rangle. \tag{A.7}$$

We denote Z_B, $VS_kV^\dagger$ and G_k acting on the subspace spanned by $\{|\Psi_{k0}\rangle, |\Psi_{k1}\rangle\}$ as $\widetilde{Z_B}$, $\widetilde{VS_kV^\dagger}$ and $\widetilde{G_k}$ respectively. The first two can be written as

$$\widetilde{Z_B} = |\Psi_{k0}\rangle \langle \Psi_{k0}| - |\Psi_{k1}\rangle \langle \Psi_{k1}|, \tag{A.8}$$

$$\begin{aligned} \widetilde{VS_kV^\dagger} = & (1 - 2\alpha_k^2) |\Psi_{k0}\rangle \langle \Psi_{k0}| \\ & + (1 - 2\beta_k^2) |\Psi_{k1}\rangle \langle \Psi_{k1}| \\ & - 2\alpha_k\beta_k(|\Psi_{k1}\rangle \langle \Psi_{k0}| + |\Psi_{k0}\rangle \langle \Psi_{k1}|). \end{aligned} \tag{A.9}$$

Therefore $\widetilde{G_k}$ is

$$\begin{aligned} \widetilde{G_k} = & \widetilde{VS_kV^\dagger}\widetilde{Z_B} \\ = & (1 - 2\alpha_k^2) |\Psi_{k0}\rangle \langle \Psi_{k0}| \\ & - (1 - 2\beta_k^2) |\Psi_{k1}\rangle \langle \Psi_{k1}| \\ & - 2\alpha_k\beta_k(|\Psi_{k1}\rangle \langle \Psi_{k0}| - |\Psi_{k0}\rangle \langle \Psi_{k1}|). \end{aligned} \tag{A.10}$$

Two eigenvalues of $\widetilde{G_k}$ are

$$\lambda_{k\pm} = e^{\pm i 2\pi \theta_k}, \tag{A.11}$$

and eigenvectors $|\Psi_{k\pm}\rangle$ each corresponding to $\lambda_{k\pm}$ are

$$|\Psi_{k\pm}\rangle = \frac{1}{\sqrt{2}}\left(|\Psi_{k0}\rangle \pm i\,|\Psi_{k1}\rangle\right). \qquad (A.12)$$

$|\Psi_k\rangle$ can be decomposed into $|\Psi_{k\pm}\rangle$ as

$$|\Psi_k\rangle = \frac{-i}{\sqrt{2}}\left(e^{i\pi\theta_k}\,|\Psi_{k+}\rangle - e^{-i\pi\theta_k}\,|\Psi_{k-}\rangle\right). \qquad (A.13)$$

[1] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
[2] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, Annu. Rev. Phys. Chem. **62**, 185 (2011).
[3] S. Aaronson, Nature Physics **11**, 291 (2015).
[4] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).
[5] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).
[6] C.-H. Yu, F. Gao, and Q.-Y. Wen, (2017), arXiv:1707.09524.
[7] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).
[8] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).
[9] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum Info. Comput. **15**, 316 (2015).
[10] S. Lloyd, M. Mohseni, and P. Rebentrost, (2013), arXiv:1307.0411.
[11] F. G. Brandao and K. M. Svore, in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2017) pp. 415–426.
[12] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, (2017), arXiv: 1710.02581.
[13] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. A **78**, 052310 (2008).
[14] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).
[15] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, Nature **471**, 87 (2011).
[16] I. Kerenidis and A. Prakash, (2016), arXiv:1603.08675.
[17] I. Kerenidis and A. Prakash, (2017), arXiv:1704.04992.
[18] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proceedings of the Royal Society A **454**, 339 (1998).
[19] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, arXiv:0005055 [quant-ph].
[20] L. Ruiz-Perez and J. C. Garcia-Escartin, Quantum Information Processing **16**, 152 (2017).
[21] F. Schmüser and D. Janzing, Phys. Rev. A **72**, 042324 (2005).
[22] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, arXiv:1711.11240.
[23] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Phys. Rev. Lett. **87**, 167902 (2001).
[24] S. S. Zhou, T. Loke, J. A. Izaac, and J. B. Wang, Quantum Inf. Process. **16**, 82 (2017).
[25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).
[26] J. A. Cortese and T. M. Braje, (2018), arXiv:1803.01958.
[27] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, arXiv:1803.00745.