



Grover search revisited: Application to image pattern matchingHiroyuki Tezuka ^{1,2,3} Kouhei Nakaji ^{2,3} Takahiko Satoh,^{2,3} and Naoki Yamamoto^{2,4,*}¹*Sony Group Corporation, 1-7-1 Konan, Minato-ku, Tokyo 108-0075, Japan*²*Quantum Computing Center, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan*³*Graduate School of Science and Technology, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan*⁴*Department of Applied Physics and Physico-Informatics, Keio University, Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan*

(Received 6 October 2021; accepted 25 February 2022; published 24 March 2022)

The landmark Grover algorithm for amplitude amplification serves as an essential subroutine in various types of quantum algorithms, with guaranteed quantum speedup in query complexity. However, there has been no proposal to realize the original motivating application of the algorithm, i.e., the database search or more broadly the pattern matching in a practical setting, mainly due to the technical difficulty in efficiently implementing the data loading and amplitude amplification processes. In this paper, we propose a quantum algorithm that approximately executes the entire Grover database search or pattern matching algorithm. The key idea is to use the recently proposed approximate amplitude encoding method on a shallow quantum circuit, together with the easily implementable inversion-test operation for realizing the projected quantum state having similarity to the query data, followed by the amplitude amplification operation that is independent to the target data index. We provide a thorough demonstration of the algorithm in the problem of image pattern matching.

DOI: [10.1103/PhysRevA.105.032440](https://doi.org/10.1103/PhysRevA.105.032440)**I. INTRODUCTION**

The Grover algorithm, or the amplitude amplification algorithm, is a landmark quantum subroutine that theoretically promises a computational advantage over any classical one, for several problems such as the satisfiability problem [1], quantum machine learning [2], the constrained polynomial binary optimization [3], and quantum amplitude estimation [4]. In this paper, we focus on the database search [5], which is in fact the first application of the Grover algorithm. The search problem is included in the following pattern matching problem. That is, we are given a set of data, i.e., a database, where each data has its own index; the task is to find out the index of a database component that has the highest similarity to a given query data. When the database contains the query data and the task is to find the corresponding unique index, then the problem is called the search. In the original Grover paper, he studied a simplified problem such that only the indices are focused, and provided the seminal amplitude amplification method that enables us to find the answer with \sqrt{n} queries while any classical one needs n queries, where n is the number of data; that is, the quadratic speedup is guaranteed.

In fact, we find many studies for implementing the original Grover algorithm. However, to our best knowledge, those are not practical ones in the above sense; that is, there has been no proposal to implement the “practical Grover algorithm” that gives the solution (index) for the search or pattern matching problem for a realistic database, with guaranteed quadratic speedup in the number of queries. This is because, in our view, there are two obstacles. The first issue is the difficulty to

prepare the quantum states of database and query. In general, to load a data vector onto a n -qubit quantum state, $O(2^n)$ quantum gates are required [6–9]. That is, the number of gates required for the data loading increases exponentially with the number of qubits, which might destroy the quantum advantage. To address this issue, sometimes the quantum random access memory (QRAM) is assumed [10,11], from which an arbitrary quantum state is loaded, but realization of QRAM seems to be difficult. The second issue, which is thought less serious than the first one, is that the operator for amplitude amplification often boils down to an “Oracle” operator. This is clearly an obstacle for the practical use of the Grover algorithm [3], because Oracle is the operator constructed with the answer to the problem; hence many previous studies treat Oracle as a black-box function [12,13].

In this paper, we propose a coherent method for realizing the Grover algorithm for search or pattern matching problems, which circumvents the above-mentioned two issues. As for the first issue, we employ the method of *approximate amplitude encoding (AAE)* that uses a constant-depth parametrized quantum circuit (PQC) for the data loading onto a quantum state [14]; in our case, we use AAE to prepare both the database state and the query state. Secondly, we formulate the amplitude amplification process such that the operators in the algorithm can be implemented without using the oracle, or in other words without knowing the index of the query state (i.e., the answer). The key of our algorithm is to use the so-called inversion-test technique to realize projection of the database state onto a subspace of states that have overlap with the query state; the amplitude of components of the projected state can then be amplified by the operator that does not contain the answer index. This framework may also be applicable to recently proposed shallow Grover algorithms [12,15], which

*yamamoto@appi.keio.ac.jp

make the diffusion operator shallower and thus preferable in noisy intermediate-scale quantum (NISQ) [16] devices. Of course, it is also beneficial for future fault-tolerant quantum computers (FTQC).

We demonstrate our algorithm in the framework of quantum image processing (QIMP) [17] in numerical simulations; particularly the error analysis of the circuit without amplitude amplification is experimentally conducted using the IBM superconducting quantum device. In QIMP, by embedding an image information onto a quantum state, we can process various tasks efficiently with much less number of bits and queues compared to the classical case. The pattern matching, which has various applications such as real-time object recognition, is one such task [18–23]. However, they did not discuss the issue of state preparation, but rather assumed that both the database and query image data are perfectly loaded onto quantum states. Moreover, most of those works employ an inefficient classical strategy that compares the query quantum state with each data quantum state in the database one by one. Clearly, we may develop the quantum strategy that prepares a single quantum state representing the whole database and further uses the Grover algorithm for amplifying the hitting probability. In fact, Ref. [19] employs the amplitude amplification operation, which is used to only enhance the above-mentioned classical strategy. Our proposed method resolves all these issues; both the query and database quantum states are prepared via AAE, and the Grover operator is constructed without knowing the target indices, for realizing the entire quantum image pattern matching algorithm applicable for a realistic image data set. In this sense, the demonstration itself is a contribution to the area of QIMP.

The rest of the paper is organized as follows. In Sec. II, we describe our enhanced pattern matching algorithm composed of the amplitude amplification (Sec. II B) and AAE (Sec. II C). Section III demonstrates the application of the algorithm to the image pattern matching problem, via the numerical simulation and the experiment with a real quantum device. Section IV concludes this paper and discusses possible directions of future research.

II. QUANTUM PATTERN MATCHING ALGORITHM

A. Basic algorithm

1. Problem setting

Our problem is described as follows. First, we have a *database* composed of N_D -dimensional real data vectors $\mathbf{a}_k = [a_{0,k}, \dots, a_{N_D-1,k}]^\top$, where $k = 0, \dots, N_I - 1$ denotes the index of data; that is, the database contains N_I data vectors. Then we are given a *query* data $\mathbf{b} = [b_0, \dots, b_{N_D-1}]^\top$, which is also an N_D -dimensional real vector. They will be encoded into quantum states and for this reason assumed to be normalized, i.e., $\sum_j a_{jk}^2 = 1$ and $\sum_j b_j^2 = 1$. Our goal is to identify the index, which we call the *target index*, of the database components that has the largest overlap (similarity) to the query \mathbf{b} ; that is, $k_* = \operatorname{argmax}_k |\mathbf{a}_k^\top \mathbf{b}|$.

2. Preparation of database and query quantum states

To execute the above-mentioned pattern matching task on a quantum device, we first need to load the data onto quantum

states. That is, we prepare the database state $|\text{database}\rangle$ and the query state $|\text{query}\rangle$ as follows:

$$|\text{database}\rangle = A|0\rangle^{\otimes(n_D+n_I)} \quad (1)$$

$$= \frac{1}{\sqrt{N_I}} \sum_{j=0}^{N_D-1} \sum_{k=0}^{N_I-1} a_{jk} |j\rangle_D |k\rangle_I \quad (2)$$

$$\equiv \frac{1}{\sqrt{N_I}} \sum_k |\text{data}(k)\rangle \otimes |k\rangle_I, \quad (3)$$

$$|\text{query}\rangle = B|0\rangle^{\otimes n_D} \quad (4)$$

$$= \sum_{j=0}^{N_D-1} b_j |j\rangle_D. \quad (5)$$

Here $\{|j\rangle_D\}$ and $\{|k\rangle_I\}$ are the orthogonal computational basis set in the data Hilbert space \mathcal{H}_D and the index Hilbert space \mathcal{H}_I , respectively. For simplicity, we assume that these spaces are identical to those of n_D and n_I qubits, meaning that $N_D = 2^{n_D}$ and $N_I = 2^{n_I}$. In Eq. (3), we define the quantum state corresponding to the k th data vector $|\text{data}(k)\rangle = \sum_{j=0}^{N_D-1} a_{jk} |j\rangle_D$; that is, the database state $|\text{database}\rangle$ is realized as a superposition of all data vectors accompanied with their indices. The database operator A and the query operator B are unitary operators for generating the corresponding quantum states. Note that, in the above expression, A and B are assumed to realize the perfect data encoding, in which case, however, an exponential number of gate operations have to be contained in the corresponding quantum circuits [6–9]. In Sec. II C, we will introduce the constant-depth circuits that approximate A and B .

Before moving to the next step, we remark that, instead of the above amplitude encoding, we can use the following basis encoded state:

$$|\text{database}\rangle = \frac{1}{\sqrt{N_I}} \sum_{j=0}^{N_D-1} \sum_{k=0}^{N_I-1} \hat{a}_{jk} |j\rangle_D |k\rangle_I, \quad (6)$$

$$|\text{query}\rangle = \frac{1}{\sqrt{N_{BE}}} \sum_{j=0}^{N_D-1} \hat{b}_j |j\rangle_D. \quad (7)$$

Here \hat{a}_{jk} and \hat{b}_j are binary (i.e., 0 or 1) variables that are determined from the original information a_{jk} and b_j . Also $\hat{\mathbf{a}}$ has N_I nonzero components and N_{BE} is the normalization constant for the query data encoded in the basis encoding. For instance, under $n_I = n_D = 2$, $\hat{\mathbf{b}} = [\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3]^\top = [0, 1, 0, 0]^\top$ if the query data is $|1\rangle_D$ and $\hat{\mathbf{a}} = [\hat{a}_{00}, \hat{a}_{10}, \hat{a}_{20}, \hat{a}_{30}, \dots, \hat{a}_{23}, \hat{a}_{33}]^\top = [0, 1, 0, 0, \dots, 0, 0]^\top$ if the database contains $|1\rangle_D$ at the index $|0\rangle_I$.

3. Basic algorithm for computing the similarity

The inner product of the classical data vectors \mathbf{a}_k and \mathbf{b} is now, in terms of quantum states, represented as the fidelity

$$\langle \text{query} | \text{data}(k) \rangle = \langle 0^{n_D} | B^\dagger | \text{data}(k) \rangle,$$

where we used the simplified notation $|0^n\rangle = |0\rangle^{\otimes n}$. This can be represented as the fidelity between

$$|\text{query}'\rangle = |0\rangle^{\otimes n} \quad (8)$$

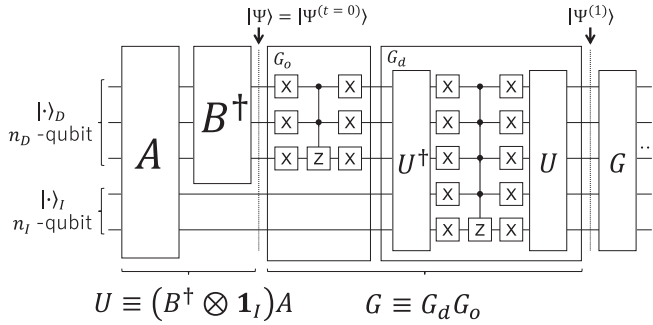


FIG. 1. Entire quantum circuit for pattern matching, composed of the AAE circuit A and B , followed by the Grover circuit G .

and $B^\dagger|\text{data}(k)\rangle$. Our algorithm uses the inversion test technique to evaluate this quantity; that is, naively, the fidelity can be computed as the probability to obtain $|\text{query}'\rangle$ when measuring the state $B^\dagger|\text{data}(k)\rangle$ in the computational basis.

Now, each $|\text{data}(k)\rangle$ is not directly given to us, but rather we have $|\text{database}\rangle = A|0\rangle^{\otimes(n_D+n_I)}$. Hence the state for the inversion test is given by

$$\begin{aligned}
 |\Psi\rangle &\equiv (B^\dagger \otimes \mathbb{1}_I)A|0\rangle^{\otimes(n_D+n_I)} \\
 &= \frac{1}{\sqrt{N_I}} \sum_k B^\dagger|\text{data}(k)\rangle \otimes |k\rangle_I,
 \end{aligned}$$

where $\mathbb{1}_I$ is the identity operator on \mathcal{H}_I . Now we make the computational-basis measurement on the first n_D qubits of $|\Psi\rangle$ and postselect the state when the result is all zeros. The entire quantum circuit for executing this task is illustrated in Fig. 1 (G is the Grover operator described later). The resultant state is given by

$$|R\rangle = \frac{1}{C_R} \sum_k r_k |k\rangle_I, \tag{9}$$

where

$$r_k = \langle \text{query} | \text{data}(k) \rangle, \quad C_R = \sqrt{\sum_k |\langle \text{query} | \text{data}(k) \rangle|^2}.$$

Thus, by further measuring the state (9) in the computational basis of \mathcal{H}_I , we are likely to obtain the index k with relatively large probability proportional to $|\langle \text{query} | \text{data}(k) \rangle|^2$, which is indeed the goal of the pattern matching task.

However, the above approach has the critical issue that the success probability of the postselection and as a result the probability of hitting the target index are severely suppressed, when the number of database components N_I is large. More precisely, the probability that the postselection succeeds and subsequently the target index k_* is identified is calculated as

$$\begin{aligned}
 P(\text{index} = k_*) &= |{}_I\langle k_* | \langle 0^{n_D} | \Psi \rangle|^2 \\
 &= \frac{1}{N_I} |\langle \text{query} | \text{data}(k_*) \rangle|^2 \\
 &\leq \frac{1}{N_I}.
 \end{aligned}$$

This means that we need $O(N_I)$ measurements for identifying the target index, which is the same computational complexity

as that of classical algorithms. This is indeed the motivation to introduce the Grover algorithm; that is, we may use the Grover algorithm to amplify the amplitude of $|0\rangle^{\otimes n_D} |k\rangle_I$ in the state $|\Psi\rangle$ for any k . The next subsection is devoted to describing the scheme in detail.

B. Amplitude amplification

First, we explain the point of the algorithm. The purpose of amplitude amplification is to amplify the amplitude of state components of $|\text{database}\rangle = A|0\rangle^{\otimes n_D+n_I}$, which have an overlap with $|\text{query}\rangle = B|0\rangle^{\otimes n_D}$. This is equivalent to amplifying the amplitude of states of $|\Psi\rangle = (B^\dagger \otimes \mathbb{1}_I)A|0\rangle^{\otimes n_D+n_I}$ that have an overlap with $|\text{query}'\rangle = |0\rangle^{\otimes n_D}$; this simple trick brings a big benefit in view of the implementation that, as shown later, the projection part (the oracle part) contained in the Grover operator is simply that on the computational basis $|0\rangle^{\otimes n_D}$ rather than that on the entangled state $|\text{query}'\rangle$. This is called the inversion test technique, which can be now employed thanks to the explicit construction of the encoding process B ; note also that the resulting Grover operator is not anymore an oracle, but is what can be explicitly implementable without knowing the target index. Once the Grover operator is constructed, it efficiently amplifies the amplitudes that are proportional to the similarity between the database component and the query, which corresponds to all nonzero r_k in Eq. (9). This is an extended framework of the conventional Grover algorithm; that is, the target states are distributed [24], rather than given as one of the basis states of the initial state. Below we apply this theory to our problem; see Appendix A for the detailed calculation.

First, the initial state $|\Psi\rangle$ and the target state $|q\rangle$ are defined as

$$|\Psi\rangle = (B^\dagger \otimes \mathbb{1}_I)A|0\rangle^{\otimes(n_D+n_I)}, \tag{10}$$

$$|q\rangle \equiv \frac{(|0^{n_D}\rangle \langle 0^{n_D}| \otimes \mathbb{1}_I) |\Psi\rangle}{\sqrt{\langle \Psi | (|0^{n_D}\rangle \langle 0^{n_D}| \otimes \mathbb{1}_I) | \Psi \rangle}}. \tag{11}$$

Recall the notation $|0^n\rangle = |0\rangle^{\otimes n}$. That is, the target state $|q\rangle$ is the component of $|\Psi\rangle$ that has an overlap with $|\text{query}'\rangle = |0\rangle^{\otimes n_D}$ in the data Hilbert space \mathcal{H}_D . More specifically, $|\Psi\rangle$ can be expressed as

$$|\Psi\rangle = \sin \theta |q\rangle + \cos \theta |q^\perp\rangle, \tag{12}$$

where $|q^\perp\rangle$ is the state orthogonal to $|q\rangle$ and $\sin \theta = \langle q | \Psi \rangle$. Also, because the state in our scenario is distributed in the basis states, it is convenient to have the expression of the above states in terms of the basis vectors as follows:

$$|\Psi\rangle = \sum_{x=0}^{N-1} \psi_x |x\rangle, \quad |q\rangle = \sum_{x=0}^{N-1} q_x |x\rangle, \tag{13}$$

where $N = N_D N_I$ is the dimension of the entire Hilbert space $\mathcal{H}_D \otimes \mathcal{H}_I$ and $\{|x\rangle\}$ is the set of basis states of this space. Then $\{b_x\}$ has at most N_I nonzero components, which can be explicitly represented as follows:

$$q_x = \begin{cases} \psi_x / \langle q | \Psi \rangle & \text{if } x \in \mathcal{C}, \\ 0 & \text{if } x \notin \mathcal{C}, \end{cases}$$

where \mathcal{C} is the set of numbers defined as

$$\mathcal{C} = \{x | \langle x | (|0^{n_D}\rangle\langle 0^{n_D}| \otimes \mathbb{1}_I) |x\rangle \neq 0\}.$$

Note that $\langle q | \Psi \rangle = [\langle \Psi | (|0^{n_D}\rangle\langle 0^{n_D}| \otimes \mathbb{1}_I) | \Psi \rangle]^{1/2}$.

The goal of amplitude amplification is to amplify the coefficient $\sin \theta$ via applying the Grover operator, which is decomposed of the following oracle operator G_o and the diffusion operator G_d :

$$\begin{aligned} |\Psi^{(t+1/2)}\rangle &= G_o |\Psi^{(t)}\rangle \\ &= (\mathbb{1} - 2|0^{n_D}\rangle\langle 0^{n_D}| \otimes \mathbb{1}_I) |\Psi^{(t)}\rangle, \\ |\Psi^{(t+1)}\rangle &= G_d |\Psi^{(t+1/2)}\rangle \\ &= U(2|0^{n_D+n_I}\rangle\langle 0^{n_D+n_I}| - \mathbb{1}_{DI}) U^\dagger |\Psi^{(t+1/2)}\rangle \\ &= (2|\Psi\rangle\langle \Psi| - \mathbb{1}_{DI}) |\Psi^{(t+1/2)}\rangle, \end{aligned}$$

where $U = (B^\dagger \otimes \mathbb{1}_I)A$ is the unitary operator producing the initial state, i.e., $|\Psi^{(0)}\rangle = |\Psi\rangle = U|0\rangle^{\otimes n_D+n_I}$. That is, the amplified state $|\Psi^{(t)}\rangle$ is generated by t applications of the Grover operator $G = G_d G_o$ on the initial state $|\Psi\rangle$. The oracle operator G_o flips the phase of the target state $|q\rangle$, while it does not change any state orthogonal to $|q\rangle$. The diffusion operator G_d inverts the amplitudes around their averaged value. Note that, although G_o is called the oracle by convention, it can be composed without any information about the target index (the answer). The entire circuit composed of the encoding and amplitude amplification operators is depicted in Fig. 1.

Following the general theory given in Appendix A, the state after t Grover operations is explicitly calculated as

$$|\Psi^{(t)}\rangle = G^t |\Psi\rangle = \sin[(2t+1)\theta] |q\rangle + \cos[(2t+1)\theta] |q^\perp\rangle,$$

showing that the amplitude of $|q\rangle$ can be amplified by appropriately choosing the number of operations. Also, corresponding to Eq. (13), the transformed state is expressed as

$$|\Psi^{(t)}\rangle = \sum_x \psi_x^{(t)} |x\rangle, \quad \psi_x^{(t)} = A_x \sin(\omega t + \delta_x),$$

where ω , A_x , and δ_x (i.e., the frequency, the amplitude, and the phase of the initial state, respectively) are given by

$$\omega = 2 \arcsin \langle q | \Psi \rangle, \quad (14)$$

$$\begin{aligned} A_x &= \sqrt{\frac{q_x^2 - 2\langle q | \Psi \rangle \psi_x q_x + \psi_x^2}{1 - \langle q | \Psi \rangle^2}} \\ &= \begin{cases} \psi_x / \langle q | \Psi \rangle & \text{if } x \in \mathcal{C}, \\ \psi_x / \sqrt{1 - \langle q | \Psi \rangle^2} & \text{if } x \notin \mathcal{C}, \end{cases} \quad (15) \end{aligned}$$

$$\begin{aligned} \delta_x &= \arccos \left(\frac{q_x - \langle q | \Psi \rangle \psi_x}{\sqrt{q_x^2 - 2\langle q | \Psi \rangle \psi_x q_x + \psi_x^2}} \right) \\ &= \begin{cases} \arccos \sqrt{1 - \langle q | \Psi \rangle^2} & \text{if } x \in \mathcal{C}, \\ \arccos(-\langle q | \Psi \rangle) & \text{if } x \notin \mathcal{C}, \end{cases} \quad (16) \end{aligned}$$

where $\langle q | \Psi \rangle = [\langle \Psi | (|0^{n_D}\rangle\langle 0^{n_D}| \otimes \mathbb{1}_I) | \Psi \rangle]^{1/2}$. Note that the phases of $\psi_x^{(t)}$ for the case $x \in \mathcal{C}$ and that for the case $x \notin \mathcal{C}$ are shifted just $\pi/2$ with each other.

Based on these results, we obtain the analytic expression of the amplified probability of hitting the matched indices; that

is, for the index $x \in \mathcal{C}$, the hitting probability after t Grover iterations is given by

$$\begin{aligned} P_x^{(t)} &= \{ \psi_x^{(t)} \}^2 \\ &= \frac{\psi_x^2}{2\langle q | \Psi \rangle^2} [1 - \cos 2(\omega t + \arccos \sqrt{1 - \langle q | \Psi \rangle^2})] \\ &\leq \frac{\psi_x^2}{\langle q | \Psi \rangle^2}. \quad (17) \end{aligned}$$

From Eq. (17), we can determine the optimal number of iteration t_* that gives the highest hitting probability P_x for any index $x \in \mathcal{C}$ as follows:

$$t_* = CI \left(\frac{\arccos \langle q | \Psi \rangle}{2 \arcsin \langle q | \Psi \rangle} \right), \quad (18)$$

where $CI(z)$ returns the closest integer of a real number z by rounding down.

The above result (18) shows that the optimal operations number t_* depends on the initial overlap $\langle q | \Psi \rangle = [\langle \Psi | (|0^{n_D}\rangle\langle 0^{n_D}| \otimes \mathbb{1}_I) | \Psi \rangle]^{1/2}$. That is, for the precise treatment of the problem, we need to estimate $\langle q | \Psi \rangle$. As indicated by this expression, this task can be conducted by estimating the success probability of projecting $|\Psi\rangle$ onto $|0\rangle^{\otimes n_D}$; but this strategy is inefficient in the sense that it needs to prepare $|\Psi\rangle$ for $O(1/\epsilon^2)$ times, where ϵ denotes the given estimation error. Instead, we could take the sophisticated amplitude estimation method [4,25,26]. With the use of this technique, the parameter ω and accordingly $\langle q | \Psi \rangle$ can be estimated via $O(1/\epsilon)$ operations of Grover. However, in a practical case where the number of database, N_I , is enough large and the database contains only few data similar to the query, we have a rough estimate $\omega \simeq \langle q | \Psi \rangle \simeq 1/\sqrt{N_I}$. In this case we obtain $t_* \simeq \sqrt{N_I}$ and further $P_x^{(t_*)} \simeq N_I \psi_x^2 \simeq 1$ if $\psi_x^2 \rightarrow 1/N_I$. This clearly means the quadratic speedup in the number of operations necessary to achieve the pattern matching, similar to the conventional Grover search algorithm.

Lastly, we remark on the implementation of the circuit. In our framework, the oracle operator G_o is simple, and it can be composed of one n_D -controlled Toffoli gate; as a result, the depth of the oracle part is $O(n_D^2)$ [27]. In contrast, the diffusion operator G_d contains U^\dagger and U , which consist of the database operator A and the query operator B . If we naively implement the perfect version of those data loading operators, the number of gates increases exponentially in the system size, i.e., $O(2^{n_D+n_I})$. This is severe to implement on NISQ, and might spoil the quantum advantages even on FTQC. To overcome this difficulty, we introduce AAE, as described in the next subsection.

C. Approximate amplitude encoding (AAE)

AAE [14] is an algorithm that trains a PQC that realizes approximate data loading. Given a target n -qubit state with real amplitudes as $|\mathbf{d}\rangle$, a PQC $U(\theta)$ with parameters θ is trained so that $U(\theta)|0\rangle^{\otimes n}$ approximates $e^{i\alpha}|\mathbf{d}\rangle$, where $e^{i\alpha}$ is the global phase. AAE runs different algorithms depending on two cases: case 1 and case 2. Case 1 is the case where the amplitudes of a target quantum state represented in the computational basis are all non-negative or nonpositive. Otherwise (i.e., for case 2), AAE offers a different algorithm. The

numerical experiment shown in Sec. III corresponds to case 1, and, therefore, we review only case 1 here.

The goal of AAE is to find $U(\theta)$ that ideally satisfies

$$e^{i\alpha}U(\theta)|0\rangle^{\otimes n} = |\mathbf{d}\rangle \equiv \sum_{j=0}^{N-1} d_j|j\rangle, \quad (19)$$

where $N = 2^n$ and d_j is the j th element of the N -element normalized real vector \mathbf{d} . To formulate a problem of finding such $U(\theta)$, we need a simple yet equivalent condition to Eq. (19); the important point is that each element of $e^{i\alpha}U(\theta)|0\rangle^{\otimes n}$ has to identify not only the absolute value but also the sign of d_j . Actually Ref. [28] proposed the method to (approximately) load only the absolute value of the coefficients by utilizing the generative adversarial network (GAN). In contrast, it was shown in [14] that Eq. (19) is equivalent to the following conditions:

$$|\langle j|U(\theta)|0\rangle^{\otimes n}|^2 = d_j^2, \quad (20)$$

$$\begin{aligned} |\langle j|H^{\otimes n}U(\theta)|0\rangle^{\otimes n}|^2 &= \left(\sum_{k=0}^{N-1} d_k \langle j|H^{\otimes n}|k\rangle \right)^2 \\ &\equiv (d_j^H)^2. \end{aligned} \quad (21)$$

Note that d_j^H is classically computable from $|\text{data}\rangle$ with complexity $O(N \log N)$, by using the Walsh-Hadamard transform [29].

The training is performed so that the following cost function \mathcal{L} is minimized by utilizing the gradient descendant algorithm:

$$\mathcal{L} = \frac{\mathcal{L}_1 + \mathcal{L}_2}{2}, \quad (22)$$

where

$$\begin{aligned} \mathcal{L}_1 &= \mathcal{L}_{\text{MMD}}(\{|\langle j|U(\theta)|0\rangle|^2\}, \{d_j^2\}), \\ \mathcal{L}_2 &= \mathcal{L}_{\text{MMD}}(\{|\langle j|H^{\otimes n}U(\theta)|0\rangle|^2\}, \{(d_j^H)^2\}). \end{aligned}$$

Here, $\mathcal{L}_{\text{MMD}}(\{q(j)\}, \{p(j)\})$ is the *maximum mean discrepancy* (MMD) between two discrete probability distributions $q(j)$ and $p(j)$ [30,31]:

$$\begin{aligned} \mathcal{L}_{\text{MMD}}(\{q(j)\}, \{p(j)\}) &\equiv \gamma_{\text{MMD}}(\{q(j)\}, \{p(j)\})^2, \\ \gamma_{\text{MMD}}(\{q(j)\}, \{p(j)\}) &= \left| \sum_{j=0}^{N-1} q(j)\Phi(j) - \sum_{j=0}^{N-1} p(j)\Phi(j) \right|, \end{aligned}$$

where $\Phi(j)$ is a function that maps the discrete random variable j to a feature space. It is shown that, as long as we choose $\Phi(k)$ so that the kernel function $\kappa(j, k) \equiv \Phi(j)\Phi(k)$ is a Gaussian kernel, the condition $\mathcal{L}_{\text{MMD}}(\{q(j)\}, \{p(j)\}) = 0$ is equivalent to $q(j) = p(j)$ for all j . In this paper, we choose Gaussian kernel functions in both \mathcal{L}_1 and \mathcal{L}_2 ; hence the condition $\mathcal{L} = 0$ is equivalent to (20) and (21), which implies the validity of minimizing the cost function \mathcal{L} .

By using AAE, the database state $|\text{database}\rangle$ and the query state $|\text{query}\rangle$ are approximately generated as follows:

$$|\text{database}\rangle \simeq A(\theta_a)|0\rangle^{\otimes n_D+n_I} \quad (23)$$

$$= \frac{1}{\sqrt{N_I}} \sum_{j=0}^{N_D-1} \sum_{k=0}^{N_I-1} \tilde{a}_{jk} |j\rangle_D |k\rangle_I \quad (24)$$

$$= \frac{1}{\sqrt{N_I}} \sum_k |\text{data}(k)\rangle \otimes |k\rangle_I, \quad (25)$$

$$|\text{query}\rangle \simeq B(\theta_b)|0\rangle^{\otimes n_D} \quad (26)$$

$$= \sum_{j=0}^{N_D-1} \tilde{b}_j |j\rangle_D. \quad (27)$$

Here, $A(\theta_a)$ and $B(\theta_b)$ are parametrized unitary operators for generating the database state and the query state, respectively; θ_a and θ_b are the optimal parameter vectors that minimize the cost function (22) for each case. Also, \tilde{a}_{jk} and \tilde{b}_j are the coefficients of the generated states by PQCs A and B , respectively. The notation \simeq in (23) and (26) represents the approximate encoding, meaning that some error may be contained. These AAE operators are followed by the Grover operator, as shown in Fig. 2, where a detailed gate structure of the AAE part is depicted.

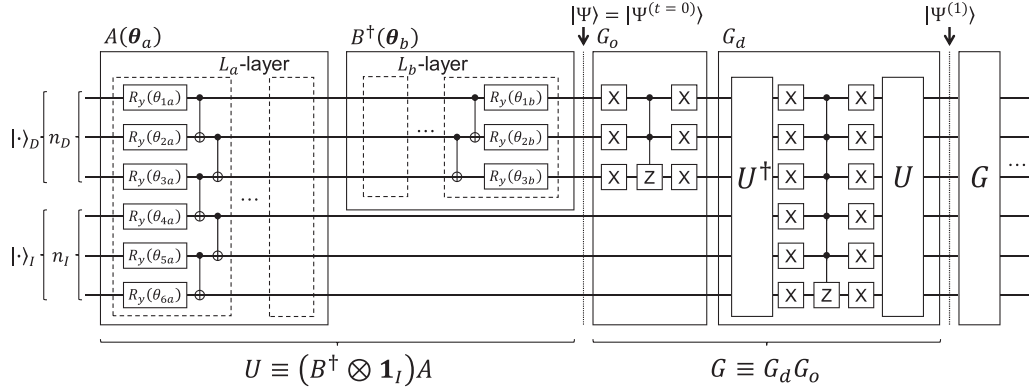
Lastly note that AAE can be applied to the basis encoded states (6) and (7) as well as the above amplitude encoded one. One may employ the exact basis encoding method using Toffoli gates, because the construction is logically straightforward and actually some efficient encoding schemes have been proposed, such as [32]. However, the total number of gates may be drastically reduced by finding an approximating encoding circuit with the use of AAE, although the learning process of the parameters will bring some encoding error. One may choose the encoding technique, the exact encoding, or AAE, depending on the situation.

III. APPLICATION TO IMAGE PATTERN MATCHING

In this section, we provide a thorough numerical demonstration of our algorithm applied to the image pattern matching problem. A set of toy image data is considered, with and without the amplitude amplification. In particular, the error analysis of the circuit for AAE and the inversion test is experimentally conducted using the IBM superconducting quantum device.

A. General quantum states for database and query image data

An image data consists of N_P pixels, where each pixel has respective color intensity represented by the integer in the range of $[0, N_C - 1]$. This means that the data vector is of $N_D = N_P N_C$ dimension. To encode this image data onto a quantum state, we take the basis encoding representation called novel enhanced quantum representation (NEQR); see Appendix B for the detailed description. Note that, as explained in Sec. II C, the basis encoding scheme can be handled via AAE.


 FIG. 2. Detailed schematic of the entire quantum circuit for pattern matching, in the case $(n_D, n_I) = (3, 3)$.

First, we take n_C and n_P qubits to represent the variable of color intensity and the pixel position, i.e., $N_C = 2^{n_C}$ and $N_P = 2^{n_P}$; then the total number of qubits of the quantum state corresponding to this image data is $n_D = n_C + n_P$. Now, let g_j be the binary-represented integer that denotes the color intensity at the j th pixel of the query data; then in the NEQR format the corresponding quantum state is assigned as

$$|\text{query}\rangle = \frac{1}{\sqrt{N_P}} \sum_{j=0}^{N_P-1} |g_j\rangle_C |j\rangle_P.$$

Note that this state can be expressed in the form (7), where \hat{b}_j takes 1 only when $|j\rangle_D$ in Eq. (7) coincides with the above $|g_j\rangle_C |j\rangle_P$. Figure 3 depicts the case where $N_P = 4$. The database quantum state is constructed in the same way. That is, denoting f_{jk} the color intensity at the j th pixel of the k th image data, the quantum correspondence is given by

$$|\text{data}(k)\rangle = \frac{1}{\sqrt{N_P}} \sum_{j=0}^{N_P-1} |f_{jk}\rangle_C |j\rangle_P.$$

This is also a $n_D = n_C + n_P$ qubit state. The database state $|\text{database}\rangle$ is the superposition of the above $|\text{data}(k)\rangle$, as defined in Eq. (3). Figure 3 depicts an example of such $|\text{database}\rangle$. Note that $\sum_{j=0}^{N_P-1} |f_{jk}\rangle_C$ and $\sum_{j=0}^{N_P-1} |g_j\rangle_C$ are normalized. Our goal is to identify the index of the elements in $|\text{database}\rangle$ that has the highest similarity with $|\text{query}\rangle$. Note that the probability to hit the index k , without the amplitude

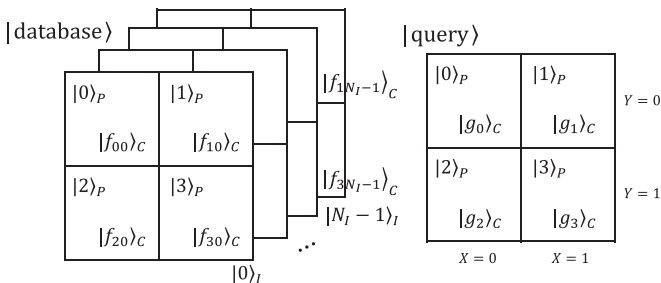


FIG. 3. Database state (left) and the query state (right).

amplification, is calculated as

$$\begin{aligned} P(\text{index} = k) &= |{}_I\langle k | \langle 0^{n_D} | \Psi \rangle|^2 \\ &= \frac{1}{N_I} |\langle \text{query} | \text{data}(k) \rangle|^2 \\ &= \frac{1}{N_I} \left[\sum_{j=0}^{N_P-1} \frac{\langle g_j | f_{jk} \rangle}{N_P} \right]^2 \\ &\leq \frac{1}{N_I}. \end{aligned} \quad (28)$$

B. Problem formulation

The query data is a four-pixel binary image data, meaning that $n_C = 1$ and $n_P = 2$. (Note that, in this binary image setting, the NEQR is equivalent to the FRQI; see Appendix B.) Then, the set of all possible data quantum state is given by $\{|0h\rangle, |1h\rangle, \dots, |Fh\rangle\}$ in the hexadecimal representation, as illustrated in Fig. 4. Here, the database is chosen as the following subset composed of eight image data (hence $n_I = 3$):

$$\begin{aligned} &\{|\text{data}(0)\rangle, |\text{data}(1)\rangle, |\text{data}(2)\rangle, |\text{data}(3)\rangle, \\ &|\text{data}(4)\rangle, |\text{data}(5)\rangle, |\text{data}(6)\rangle, |\text{data}(7)\rangle\} \\ &= \{|0h\rangle, |2h\rangle, |4h\rangle, |6h\rangle, |8h\rangle, |Ah\rangle, |Ch\rangle, |Eh\rangle\}. \end{aligned}$$

Therefore, the database quantum state $|\text{database}\rangle$ is a six-qubit quantum state, composed of $n_C = 1$ qubit for the color intensity, $n_P = 2$ qubits for the pixel position ($X \in \{0, 1\}$ and $Y \in \{0, 1\}$), and $n_I = 3$ qubits for the index.

In this demonstration, we consider the following three different settings: (i) QASM simulator on Qiskit [33] is used to simulate AAE and the pattern matching part, (ii) the optimal parameters θ_a and θ_b obtained in the setup (i) are used to run the circuit for pattern matching on the superconducting

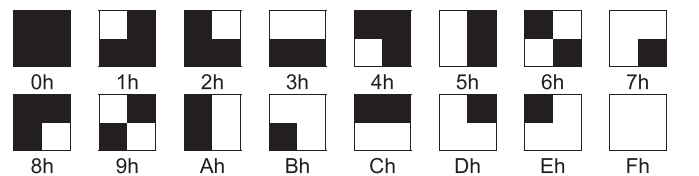


FIG. 4. Four pixel-binary data. “h” means the hexadecimal representation.

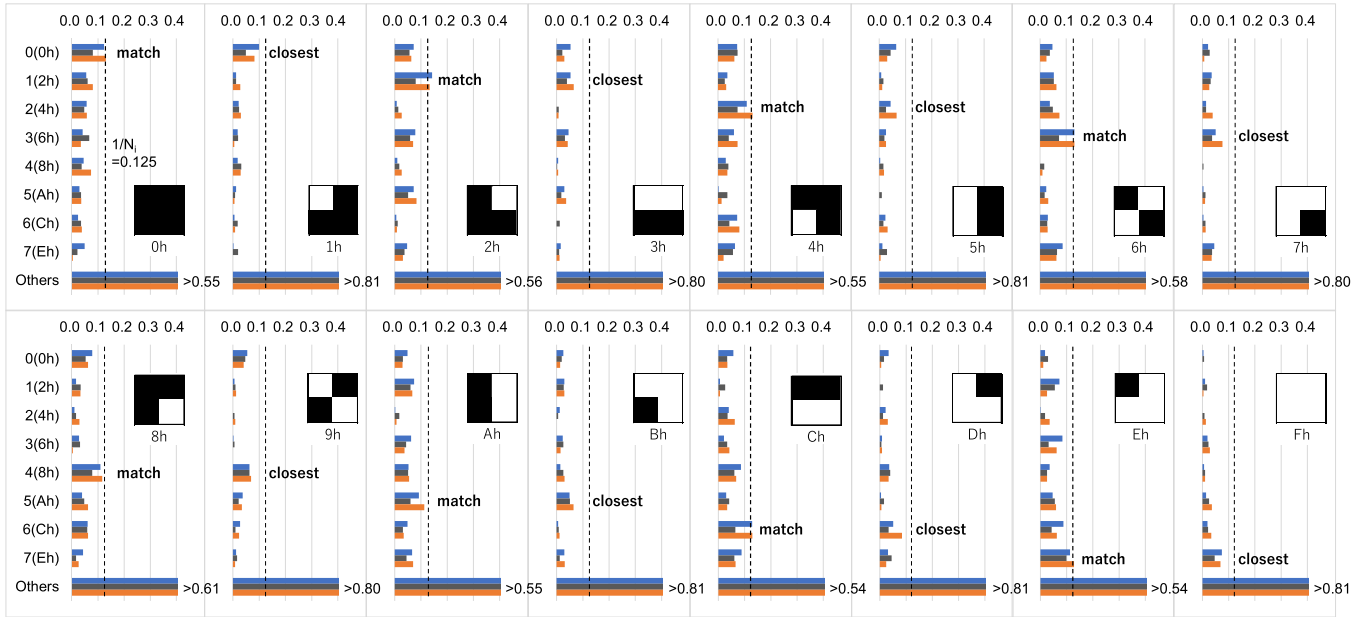


FIG. 5. Result of the pattern matching test without amplitude amplification. The vertical axis is the index (data) of the database (index $0, 1, 2, 3, 4, 5, 6, 7$) = (data 0h, 2h, 4h, 6h, 8h, Ah, Ch, Eh). The bars represent the probability $P(\text{index} = k) = |\langle k | \langle 0^{nd} | \Psi \rangle|^2$; the blue (upper), gray (middle), and orange (lower) bars correspond to the cases (i), (ii), and (iii), respectively. “Others” in the vertical axis means the failure probability, i.e., the total probability of projection of $|\Psi\rangle$ onto the states other than $|0\rangle^{\otimes nd}$. “Match” represents the index of the database component that coincides with the query with highest probability. “Closest” represents the index of the database component that does not exactly match but is the closest to the query. The square images in each figure are the query data.

quantum processor (ibm_kawasaki), and (iii) the data encoding process is executed via the perfect encoding algorithm containing Toffoli gates, rather than AAE; the pattern matching process is then operated using the QASM simulator.

We executed AAE to encode the image data as follows. The PQC for encoding the database state, $A(\theta_a)$, is a six-layers hardware efficient ansatz (HEA) and that for the query state, $B(\theta_b)$, is a three-layers HEA. Each layer is composed of the parametrized single-qubit Y -rotational gate $R_y(\theta_r) = \exp(-i\theta_r \sigma_y/2)$ and controlled-NOT (CNOT) gate that connect adjacent qubits, as shown in Fig. 2, where θ_r is the r th parameter and σ_y is the Pauli Y operator [hence $A(\theta_a)$ and $B(\theta_b)$ are real matrices]. We randomly initialized all θ_r , at the beginning of each training. As for the kernel function, $\kappa(x, y) = \exp[-64(x - y)^2]$ is used. To compute the r th gradient of the loss function, we generate 400 samples for each q_{θ}^+ , q_{θ}^- , q_{θ}^{H+} , and q_{θ}^{H-} for training $B(\theta_b)$ and 10 000 samples for the case of $A(\theta_a)$. As the optimizer, Adam [34] is used; the learning rate is 0.1 for the first 100 epochs and 0.01 for the other epochs. The number of iterations (i.e., the number of the updates of the parameters) for training the PQC is set to 300 for $B(\theta_b)$ and 500 for $A(\theta_a)$.

After the training process of AAE, we run the pattern matching algorithm. Both with and without the amplitude amplification via Grover operation, we construct the probability distribution $P(\text{index} = k)$, for all $k \in \{0, 1, \dots, 7\}$ in Eq. (28), to estimate the set of indices of the data that has significant similarity to the query. The number of samples (or the shot) is chosen depending on a specified precision, but it is 512 for all the results displayed in what follows.

C. Results and discussion

The probability distribution of the index variable, without amplitude amplification (i.e., $t = 0$), is shown in Fig. 5; that is, the empirical probability value of $P(\text{index} = k) = |\langle k | \langle 0^{nd} | \Psi \rangle|^2$ given in Eq. (28) is calculated by sampling and then horizontally displayed, while the indices of the database component (index: $0, 1, 2, 3, 4, 5, 6, 7$) = (data: 0h, 2h, 4h, 6h, 8h, Ah, Ch, Eh) are shown in the vertical axis. “Others” in the vertical axis means the failure probability, i.e., the total probability of projection of $|\Psi\rangle$ onto the states other than $|0\rangle^{\otimes nd}$. The blue (upper), gray (middle), and orange (lower) bar corresponds to the cases (i), (ii), and (iii), respectively; see Sec. III B for the meaning of these cases. Each subfigure shows the result with respect to the different query data, e.g., 0h in the top left.

This result shows that the algorithm works well; for both simulation (i) and the experiment (ii), the AAE followed by the inversion test circuit produces the outcomes that agree well with the theoretical prediction (iii). In particular, the database component obtained with highest probability actually coincides with the query, as indicated by “match” in the figure. This makes sense because, as shown in Eq. (28), the probability value with respect to the index corresponds to the overlap, or the fidelity, between the database component and the query data, which has the maximum value $1/N_l$. Note that those probability values reflect the Hamming distance between two images in this case. Hence, as a practical use case, a user may set a threshold value of the similarity score based on the Hamming distance and apply the result to identify some candidate patterns that have the above-threshold overlap with

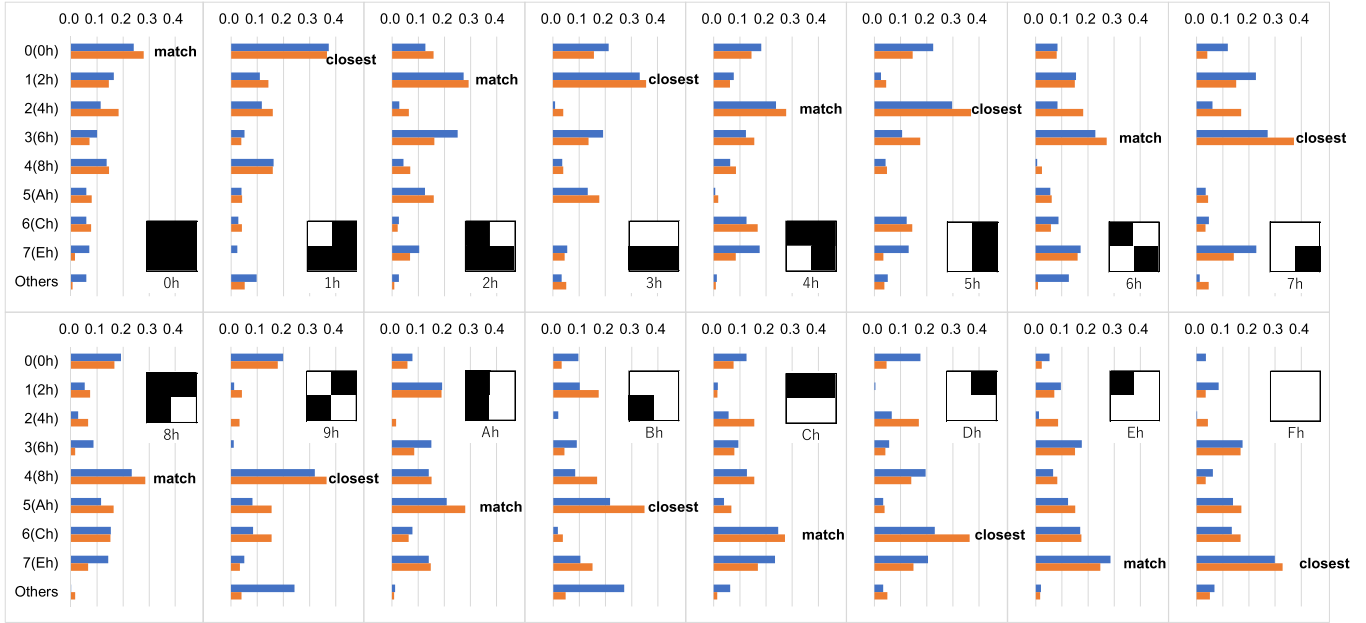


FIG. 6. Result of the pattern matching test using amplitude amplification, with the number of Grover iterations $t = 5$. The blue (upper) and orange (lower) bars correspond to the cases (i) and (iii), respectively. “Match” represents the index of the database component that coincides with the query with highest probability. “Closest” represents the index of the database component that does not exactly match but is the closest to the query. Note that the oracle operator is the same in all cases.

the query, e.g., (0h, 2h, 6h, Ah) for the query 2h. Moreover, even if the database does not contain exactly the same data as the query (hence there is no “match” index), this idea of posing the candidates works; “closest” in the figure represents the index of database component that is not exactly a match with but the closest to the query. Note that, though the experimental result (ii) agrees with the theory (iii) as mentioned above, we can clearly see that the quantum noise (decoherence) degrades the performance of the encoding performance, as shown in Fig. 5. That is, many of the hitting probabilities of the database states decrease and the probability of “others” (the failure probability) becomes higher, compared to the case of the ideal simulation result (i). This total degradation of the hitting probabilities is presumably due to the depolarization noise, which uniformly decreases the amplitudes of state.

We now turn our attention to the case with amplitude amplification, under the same setting as above. We take the number of Grover iterations to be $t = 5$. The result is shown in Fig. 6, where the quantities in the graph have the same meaning as those in Fig. 5, although in this case we have not conducted the experiment with a real quantum device. Clearly, the amplitude amplification works well, since the failure probability of “Others” is drastically reduced and, equivalently, the success probability of the postselection is effectively enhanced. As a result, all success probabilities are amplified, while keeping the relative ratio due to Eq. (17) showing that all the coefficients ψ_x are scaled with the same factor of $1/\langle q|\Psi\rangle$. Overall, the result obtained with the use of AAE (represented by the blue upper bars) and that with the ideal encoder (orange lower bars) show good agreement, but some deviations are observed. This is mainly because of the sampling and encoding errors of AAE. In fact, AAE is a machine learning technique to find the approximate encoder,

so a nonzero error is in general inevitable. The impact of such errors on our pattern matching algorithm will be investigated in another example, demonstrated in Appendix B 3.

Let us now discuss the relation between the number of Grover iterations, t , and the probability (17) to hit the index k of the database component, for the case where the query data is 0h or 1h. The result is plotted in Figs. 7 and 8. Clearly, the scaling factor and the frequency of magnification ratio are different depending on the query data, reflecting the term $\langle q|\Psi\rangle$. As mentioned in Sec. II B, to exactly specify the optimal number of iteration t_* , we need to estimate the value of $\langle q|\Psi\rangle$. This task seems to be severe when the frequency of the oscillation

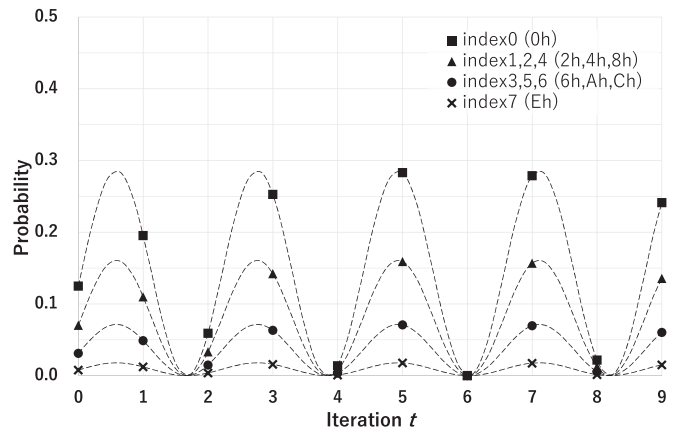


FIG. 7. Amplified probability versus the number of iterations. The plots represent the result obtained by the numerical simulation with ideal encoding scheme. The dashed lines represent the analytical expression (17). The database is shown in Fig. 5 and the query data is “0h.”

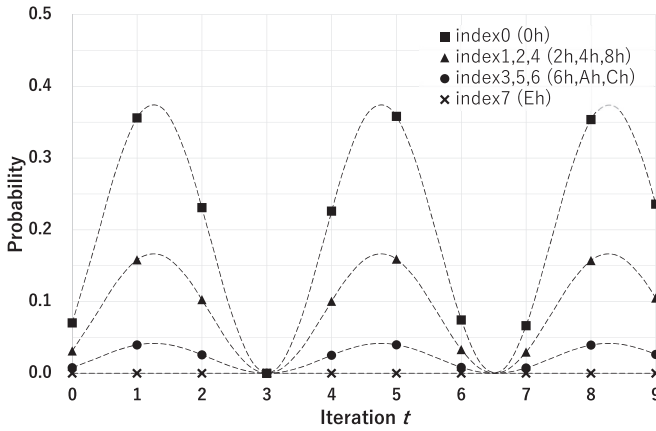


FIG. 8. Amplified probability versus the number of iterations. All lines are generated in the same way explained in Fig. 7. The query data is taken here as “1h.”

$\omega = 2 \arcsin \langle q|\Psi \rangle$ given in Eq. (14) is high. However, for a practical database where N_I is large, ω becomes small and as a result the probability (17) becomes a monotonically increasing function with respect to t , meaning that a rough choice of t might work to amplify the target probability. Another notable point is that eight indices are categorized to four groups. These groups reflect the Hamming distance (HD) between the query data and each database component, namely $HD = 0,1,2,3$ in Fig. 7 and $HD = 1,2,3,4$ in Fig. 8, from top to bottom. This result shows that our algorithm is quite natural in the context of pattern matching, although this clear relationship is due to the fact that the data are given by the binary images.

D. Depth and the number of multiqubit gate

Here we discuss the size of the circuit for implementing our algorithm. First recall that AAE drastically reduces the circuit depth and the number of multiqubit gates for implementing a data loading circuit, compared to the conventional encoding scheme. Actually, in the previous subsection, we demonstrated that the six-qubits database quantum state can be well prepared with a six-layer HEA, which contains 30 CNOT gates. This is in stark contrast to the conventional exact encoding circuit that requires 32 six-qubit Toffoli gates, which is decomposed to 128 CNOT gates and 96 auxiliary qubits [35]. One can take a more efficient method [32] that allows us to generate an arbitrary n -qubit state from $|0 \dots 0\rangle$ using $2^n - n - 1$ CNOT gates (including long-range gates), i.e., at most 57 CNOT gates for the case $n = 6$. However, if we are allowed to use only the nearest-neighbor interaction of qubits like the current superconducting quantum devices, the number of CNOT gates is estimated as follows [35]:

$$\frac{10}{3}2^n + 2n^2 - 12n + \begin{cases} 14/3, & n \text{ even,} \\ 10/3, & n \text{ odd,} \end{cases}$$

meaning that at most there are 218 CNOT gates in the case $n = 6$. Summarizing, the above exact encoding methods need an exponential number of CNOT gates, which is quite challenging.

In contrast, with PQC, the depth and the number of multiqubit gates seem not to increase so fast, i.e., $O(\text{poly}(n))$ in depth and $O(n)$ of multiqubit gates for n -qubits systems,

although in general a PQC does not guarantee a precise state preparation. Therefore, we focus on problems that do not require a highly precise encoding of the database; in fact, the pattern matching problem needs only a relative fidelity between the query data and the database components, and in this case a shallow encoding circuit may still work as demonstrated in the simulation. In this direction, though in this paper we took a simple HEA, it is important to develop a designing theory of an $O(\text{poly}(n))$ depth PQC that can prepare the database state with the necessary precision. For instance, we may incorporate the database structure to the ansatz design, to which some existing techniques (see, e.g., Refs. [36,37]) could be applied. Also, care should be taken to mitigate the so-called gradient vanishing (barren plateau) issue [38] for designing the variational ansatz; we could apply some proposals such as the circuit initialization technique [39], use of a special structured ansatz [40], and the parameter embedding method [41].

Lastly, in our demonstration we naively implemented the diffusion operator G_d , but there are several implementation schemes which reduce the number of elementary gates and the circuit depth to compose the diffusion operator [12,15,42], which were recently demonstrated [43–45]. Application of those methods will enable us to demonstrate the amplitude amplification process executed on a real quantum device.

IV. CONCLUSIONS

In this paper, we proposed a method for approximately executing the database search or more broadly the pattern matching algorithm, i.e., the algorithm that Grover originally considered as a practical application of quantum computation. The key idea is to implement the data loading process on a shallow parametrized quantum circuit and the pattern matching function on the inversion-test based circuit, followed by the amplitude amplification operation that can be constructed without using the target index. The data-loading circuit needs much fewer multiqubit entangling gates than the conventional one; introducing the recent technique to implement the diffusion operator with small blocks [15] may reduce the difficulty to implement the amplitude amplification part as well. Our proposed framework will then be totally beneficial for NISQ but also for FTQC devices. The demonstration of algorithm in the problem of image pattern matching, with both numerical simulations and partially a real quantum device, contributes to the field of quantum image processing.

Lastly we discuss the possible quantum advantage of the proposed method, where of course the computational overhead in the variational part of AAE should be carefully taken into account. First, note that the computational complexity in the entire algorithm need not include the cost for preparing the database quantum state as in the classical case. Therefore, the core of complexity lies in the overhead to prepare the query quantum state; if the number of repetition of the AAE circuit $B(\theta)$ becomes dominant over the number of Grover operations, the quantum advantage will vanish. Thus it is important to limit each data vector to a low-dimensional one such as that of a telephone number or a low-resolution image used for object recognition, which can be represented with, e.g., up to six qubits. In this case we expect that AAE would take

roughly hundreds of iterations of $B(\theta)$ to encode the query data with enough precision, or even the quantum circuit for exact encoding could be implementable nearly perfectly. The point is that, in this scenario, the complexity for preparing the quantum query data state does not scale with the number of database, N_I , which is usually quite big; the size of a phone book is much bigger than the dimension of each data of phone number. Hence our view is that, when $N_D \ll N_I$ and a very precise preparation of the database state is not required, the proposed algorithm for search or pattern match may have a practical quantum advantage both in memory and query complexities. Surely the hardness of near-perfect implementation of the Grover operator remains, which yet might be attacked via several recent proposals [12,15] together with possible circumvention via exploiting the specific structure of A and B . These important problems, along with customizing our framework for other applications, will be presented elsewhere.

ACKNOWLEDGMENTS

This work was supported by MEXT Quantum Leap Flagship Program Grants No. JPMXS0118067285 and No. JPMXS0120319794.

APPENDIX A: GROVER ALGORITHM

Here we show the Grover algorithm, in a generalized form where the target state (the query state in our language) is not limited to an eigenstate of the initial state. See, e.g., Ref. [24] for a more detailed description.

We first define an initial state $|\Psi\rangle$ as follows:

$$\begin{aligned} |\Psi\rangle &\equiv \sin\theta|q\rangle + \cos\theta|q^\perp\rangle, \\ |q\rangle &\equiv \frac{1}{N}P|\Psi\rangle, \\ |q^\perp\rangle &\equiv \frac{1}{N'}(\mathbb{1} - P)|\Psi\rangle. \end{aligned} \quad (\text{A1})$$

That is, $|\Psi\rangle$ is decomposed to a given target state $|q\rangle$ and its orthogonal state $|q^\perp\rangle$, where $N = \sqrt{\langle\Psi|P|\Psi\rangle}$ and $N' = \sqrt{\langle\Psi|(\mathbb{1} - P)|\Psi\rangle}$ are the normalization constants. P is a projection matrix, which defines the component(s) of $|\Psi\rangle$ to be searched. Also note that $\sin\theta = \langle q|\Psi\rangle$. The goal is to amplify the coefficient of the target state, i.e., $\sin\theta$, via acting on the Grover operator

$$G \equiv G_d G_o,$$

where G_o and G_d are the oracle operator and the diffusion operator defined as

$$\begin{aligned} G_o &\equiv \mathbb{1} - 2P, \\ G_d &\equiv 2|\Psi\rangle\langle\Psi| - \mathbb{1}. \end{aligned}$$

Actually, by combining the above equations, we have

$$G \begin{bmatrix} |q\rangle \\ |q^\perp\rangle \end{bmatrix} = \begin{bmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{bmatrix} \begin{bmatrix} |q\rangle \\ |q^\perp\rangle \end{bmatrix}.$$

This represents that G induces a rotation by 2θ in the space spanned by $|q\rangle$ and $|q^\perp\rangle$. As a result, we found that $|\Psi^{(t)}\rangle \equiv G^t|\Psi\rangle$ is given by

$$|\Psi^{(t)}\rangle = \sin[(2t+1)\theta]|q\rangle + \cos[(2t+1)\theta]|q^\perp\rangle. \quad (\text{A2})$$

Hence certainly the amplitude of $|q\rangle$ is amplified, by appropriately choosing the number of operations.

Next, let us express $|\Psi\rangle$ and $|q\rangle$ in terms of the basis states $\{|x\rangle\}$ as follows:

$$|\Psi\rangle \equiv \sum_x \psi_x |x\rangle, \quad (\text{A3})$$

$$|q\rangle \equiv \sum_x q_x |x\rangle. \quad (\text{A4})$$

For simplicity, we assume ψ_x and q_x are real numbers for all x . Correspondingly, $|\Psi^{(t)}\rangle$ can also be expressed using the basis states $\{|x\rangle\}$:

$$|\Psi^{(t)}\rangle \equiv \sum_x \psi_x^{(t)} |x\rangle.$$

We now derive the explicit representation of $\psi_x^{(t)}$, to see how each coefficient ψ_x of $|\Psi\rangle$ is amplified. From Eqs. (A1), (A3), and (A4), we have

$$\begin{aligned} |q^\perp\rangle &= \frac{1}{\cos\theta}|\Psi\rangle - \frac{\sin\theta}{\cos\theta}|q\rangle \\ &= \sum_x \left[\frac{\psi_x}{\cos\theta} - \frac{\sin\theta}{\cos\theta}q_x \right] |x\rangle \\ &\equiv \sum_x q_x^\perp |x\rangle. \end{aligned} \quad (\text{A5})$$

Then, by substituting Eqs. (A4) and (A5) into Eq. (A2), we have

$$|\Psi^{(t)}\rangle = \sum_x [\sin[(2t+1)\theta]q_x + \cos[(2t+1)\theta]q_x^\perp] |x\rangle.$$

Therefore,

$$\begin{aligned} \psi_x^{(t)} &= \sin[(2t+1)\theta]q_x + \cos[(2t+1)\theta] \frac{\psi_x - q_x \sin\theta}{\cos\theta} \\ &= \frac{1}{\cos\theta} [q_x \sin(2t\theta) + \psi_x \cos(2t\theta + \theta)] \\ &= \frac{1}{\cos\theta} [(q_x - \psi_x \sin\theta) \sin(2t\theta) + \psi_x \cos\theta \cos(2t\theta)] \\ &= \sqrt{\frac{\psi_x^2 - 2\langle q|\Psi\rangle\psi_x q_x + q_x^2}{1 - \langle q|\Psi\rangle^2}} \sin(2t\theta + \phi_x), \end{aligned} \quad (\text{A6})$$

where

$$\theta = \arcsin\langle q|\Psi\rangle,$$

$$\phi_x = \arccos\left(\frac{q_x - \langle q|\Psi\rangle\psi_x}{\sqrt{\psi_x^2 - 2\langle q|\Psi\rangle\psi_x q_x + q_x^2}}\right).$$

Hence the phase ϕ_x determines the amplification gain as a function of x . Note that, in the original Grover algorithm, $|\Psi\rangle$ is the superposition of equiprobable basis states, i.e., $\psi_0 = \dots = \psi_{N-1} = 1/\sqrt{N}$, and the target state is one of them, meaning that $P = |y\rangle\langle y|$ and accordingly $|q\rangle = |y\rangle$ and $q_x = \delta_{x,y}$. In this case, $\sin\theta = \langle q|\Psi\rangle = 1/\sqrt{N}$, which leads to

$$\psi_y^{(t)} = \sin(2t\theta + \phi_y), \quad \phi_y = \arccos\sqrt{1 - 1/N} = \theta,$$

and for $x \neq y$

$$\psi_x^{(t)} = \frac{\sin(2t\theta + \phi_x)}{\sqrt{N-1}}, \quad \phi_x = \arccos\frac{1}{\sqrt{N}} = \theta + \frac{\pi}{2}.$$

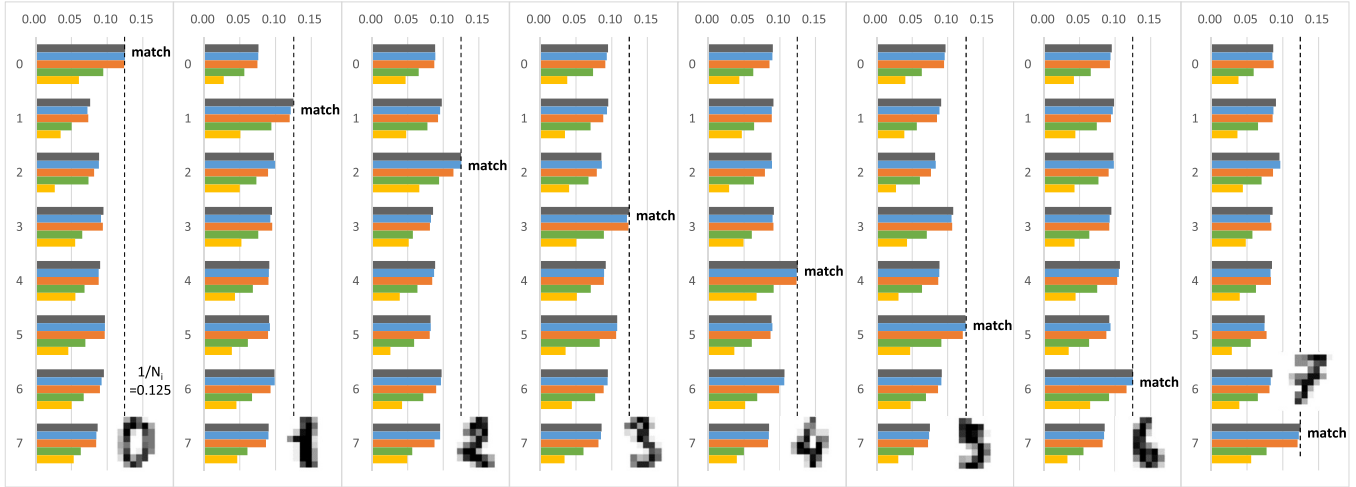


FIG. 9. Result of the pattern matching test when using FRQI. The vertical axis is index (data) of the database (index 0,1,2,3,4,5,6,7) = (digit 0,1,2,3,4,5,6,7). The horizontal axis is the probability $P_{\text{FRQI}}(\text{index} = x)$. “Match” represents the index of database component that coincides with the query with highest probability. The gray (top) bar is the case of ideal encoding. The blue, orange, green, and yellow (the second from top to bottom) bars are the cases where the encoding noise are 5%, 10%, 30%, and 50% (corresponding fidelity values are 0.99, 0.98, 0.86, and 0.72, respectively). The single data is represented using seven qubits (one qubit for the color intensity and the others for position) and the database is represented using 10 qubits since it has eight indices.

This coincides with the formula given in the original Grover algorithm [46].

In our context, the initial state is $|\Psi\rangle = \sum_x \psi_x|x\rangle$ with $\{|x\rangle\}$ the set of basis states of $\mathcal{H}_D \otimes \mathcal{H}_I$. The target state $|q\rangle$ is characterized by the projection $P = |0^{nd}\rangle\langle 0^{nd}| \otimes \mathbb{1}_I$; that is, $|q\rangle$ is the state having the query data state $|\text{query}'\rangle = |0\rangle^{\otimes nd}$ for any index component. Then we have

$$|q\rangle = \frac{1}{N} \sum_x \psi_x P|x\rangle = \frac{1}{N} \sum_{x \in \mathcal{C}} \psi_x|x\rangle,$$

where \mathcal{C} is the set of numbers defined as

$$\mathcal{C} = \{x|\langle x|(|0^{nd}\rangle\langle 0^{nd}| \otimes \mathbb{1}_I)|x\rangle \neq 0\}.$$

Note that $N = \langle q|\Psi\rangle = [(\Psi|(|0^{nd}\rangle\langle 0^{nd}| \otimes \mathbb{1}_I)|\Psi)]^{1/2}$. As a result,

$$q_x = \begin{cases} \psi_x/\langle q|\Psi\rangle & \text{if } x \in \mathcal{C}, \\ 0 & \text{if } x \notin \mathcal{C}. \end{cases}$$

Then the state after t Grover operations, $|\Psi^{(t)}\rangle = G^t|\Psi\rangle = \sum_x \psi_x^{(t)}|x\rangle$, can be specified as follows: that is, from Eq. (A6), we find that $\psi_x^{(t)} = A_x \sin(\omega t + \delta_x)$, where $\omega = 2\theta$, A_x , and δ_x are given by Eqs. (14), (15), and (16), respectively. Also the hitting probability $P_x^{(t)} = \{\psi_x^{(t)}\}^2$ can also be readily calculated and given by Eq. (17).

APPENDIX B: QUANTUM IMAGE REPRESENTATIONS

Various quantum image representation frameworks have been proposed [47]. They are categorized to the amplitude-encoding-based format and the basis-encoding-based format. The basic idea of the former is summarized by flexible representation for quantum images (FRQI) and the latter is summarized by novel enhanced quantum representation (NEQR). In this section, these two representations are reviewed; note that both representations can be taken in our proposed algorithm.

1. Flexible representation for quantum images (FRQI)

In general, quantum image representation is a method for expressing the pixel positions and the corresponding color intensities, as a quantum superposition state. In the FRQI format [48], a grayscale image on $N_p = 2^n \times 2^n$ pixels is encoded in a quantum state as follows:

$$\begin{aligned} |I_{\text{FRQI}}\rangle &= \frac{1}{\sqrt{N_p}} \sum_{z=0}^{N_p-1} (\cos \theta_z|0\rangle + \sin \theta_z|1\rangle)|z\rangle \\ &= \frac{1}{\sqrt{N_p}} \sum_{z=0}^{N_p-1} |f_z\rangle|z\rangle, \end{aligned} \tag{B1}$$

where $|0\rangle$ and $|1\rangle$ are the computational basis states of a single qubit, while $|z\rangle$ is the computational basis state of $\log N_p = 2n$ qubits representing the coordinate of the corresponding pixel. $\theta = (\theta_0, \theta_1, \dots, \theta_{2n-1})$, $\theta_i \in [0, \pi/2]$ is the vector of angles encoding the colors; that is, $|f_z\rangle = \cos \theta_z|0\rangle + \sin \theta_z|1\rangle$ is a qubit on which the color at the pixel coordinate z is encoded. That is, a grayscale $2^n \times 2^n$ pixel image is represented by the $2n + 1$ qubit state (B1).

The point of FRQI is that it needs only the minimum number of qubits in which one can encode the absolute value of color information with arbitrary precision. This is advantageous to amplitude encoding frameworks such as [49], where the color information is directly encoded into the amplitude of state vectors representing the pixel position. Actually, the scheme [49] cannot represent the absolute color value but only the relative value, in contrast to FRQI and NEQR; this induces an undesirable change of amplitude information depending on the other pixel’s one, which happens even if the absolute intensities are the same, e.g., in the case where only one pixel has maximum intensity and others are zero or the case where two pixels have the maximum intensity.

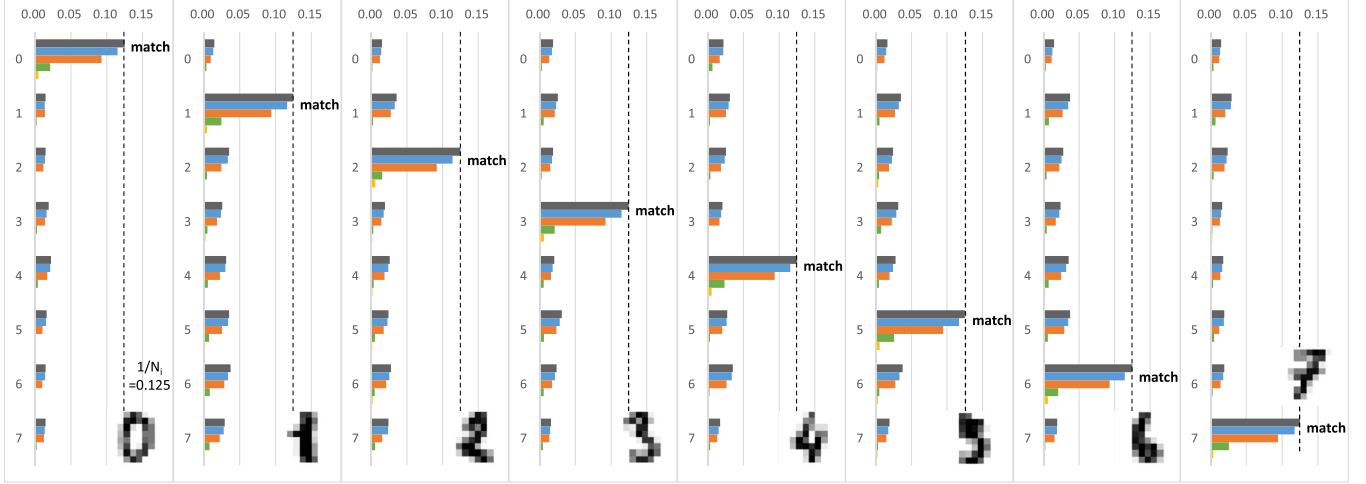


FIG. 10. Result of the pattern matching test when using NEQR. The vertical axis is index (data) of the database (index 0,1,2,3,4,5,6,7) = (digit 0,1,2,3,4,5,6,7). The horizontal axis is the probability $P_{\text{NEQR}}(\text{index} = x)$. The gray (top) bar is the case of ideal encoding. The blue, orange, green, and yellow (the second from top to bottom) bars are the cases where the encoding noise are 5%, 10%, 30%, and 50% (corresponding fidelity values are 0.96, 0.86, 0.41, and 0.20, respectively). The single data is represented using 10 qubits [four qubits for the color intensity (16 levels) and the others for position] and the database is represented using 13 qubits since it has eight indices.

On the other hand, FRQI has the following disadvantage: that is, an exponential number of measurement is needed to restore the exact image since one has to identify all the state amplitudes. However, if we are interested only in the intermediate process rather than retrieving the complete image information, this measurement issue is not crucial, as pointed out in [17]. Actually, our pattern matching problem does not need such an exact information retrieval, and the user can adjust the sampling cost only depending on the required accuracy.

Finally, in this FRQI representation, the probability distribution with respect to the index variable, Eq. (28), is explicitly given by

$$P_{\text{FRQI}}(\text{index} = x) = \frac{1}{N_I} \left(\frac{N_p^{\text{FRQI}}(x)}{N_p} \right)^2, \quad (B2)$$

$$N_p^{\text{FRQI}}(x) = \sum_{j=0}^{N_p-1} (\cos \theta_j \cos \theta_{jx} + \sin \theta_j \sin \theta_{jx}),$$

where θ_j and θ_{jx} correspond to the color intensity of the j th pixel in the query data and the color intensity of the j th pixel of the x th image in the database, respectively.

2. Novel enhanced quantum representation (NEQR)

NEQR proposed in [50] is used for encoding a digital image, meaning that the color intensity is represented by a binary code. In this scheme, the color information represented by the discrete variable taking the 2^q values can be stored in a q -qubit system. If the image is created on $N_p = 2^n \times 2^n$ pixels, it is encoded into the quantum state of the form

$$|I_{\text{NEQR}}\rangle = \frac{1}{\sqrt{N_p}} \sum_{z=0}^{N_p-1} \otimes_{i=0}^{q-1} |c_z^i\rangle |z\rangle$$

$$= \frac{1}{\sqrt{N_p}} \sum_{z=0}^{N_p-1} |f(z)\rangle |z\rangle, \quad (B3)$$

where $c_z^i \in \{0, 1\}$ is a bit information and $f(z) = \otimes_{i=0}^{q-1} c_z^i \in [0, 2^q - 1]$ is a bit string representing the pixel color information at the coordinate z . Clearly, the total number of qubit depends on the level of quantization of the color information, and fine encoding needs much more qubits than FRQI. On the other hand, one can restore an N_p -pixel quantum image exactly with $O(N_p)$ measurement, thanks to the orthogonality of the states. In this NEQR representation, the probability distribution with respect to the index variable, Eq. (28), is given by

$$P_{\text{NEQR}}(\text{index} = x) = \frac{1}{N_I} \left(\frac{N_p^{\text{NEQR}}(x)}{N_p} \right)^2,$$

$$N_p^{\text{NEQR}}(x) = \sum_{j=0}^{N_p-1} \otimes_{i=0}^{q-1} \langle c_j^i | c_{jx}^i \rangle, \quad (B4)$$

where c_j^i and c_{jx}^i represent the i th bit of the color intensity bit string of the j th pixel in the query data and the i th bit of the color intensity bit string of the j th pixel of the x th image in the database, respectively.

3. Impact of the encoding error for the case of handwritten digit data

Here we show the result of the analysis on the impact of the encoding error, depending on the encoding schemes; that is, we will display the two probability distributions (B2) and (B4) under some errors. We use the ‘‘Handwritten Digits’’ data set [51] composed of 10 handwritten digits from numbers 0 to 9; each data is an 8×8 pixels image with 16 level color assigned by the amplitude variable $u_x \in [0, 1]$, where x denotes the position index of the image. We suppose that the amplitude u_x is affected by the noise ϵ_x generated from the normal distribution $N(0, \sigma)$, $\sigma = \max(\{u_x\})\sigma_0$. Then we make the normalized amplitude vector $\{u_x + \epsilon_x | x \in \text{all basis}\}$ divided by $\sqrt{\sum_x (u_x + \epsilon_x)^2}$. The above procedures are conducted for

all the database components and a query data. Then we calculate the probability distributions (B2) and (B4) for several noise deviations $\sigma_0 \in \{0.05, 0.1, 0.3, 0.5\}$ to see the impact of noise. Note that we consider the case without amplitude amplification. In the case with amplitude amplification, the amplitudes are magnified the factor of $1/\langle q|\Psi\rangle$ while keeping the relative amplitude.

Results are shown in Figs. 9 and 10. Both FRQI and NEQR work appropriately, giving the correct indices with highest score. In terms of distinguishability, NEQR seems better, because of the clear difference between the matched state and the others. However, this also means that we cannot see well the similarity degree between each component. In other words, FRQI is appropriate for fuzzy matching and NEQR for exact matching. For example, we can see in Fig. 9 that “3” and “5” are similar, since those scores are higher than others (Index 5 in Query 3 and Index 3 in Query 5). Thus one should choose the data format depending on the purpose.

In terms of error tolerance, FRQI is better. When the noise magnitude is less than 10%, it does not largely affect the similarity score in FRQI. On the other hand, in NEQR, some non-negligible impact is observed even with 5% error and, moreover, the score drastically becomes small with 30% error. These behaviors are explained by their representation formats. In FRQI, the color intensity is expressed by a single qubit, $\cos\theta|0\rangle + \sin\theta|1\rangle$, which is independent to the color-level precision, and most substates have nonzero amplitude with ordinary images (highlight or shadow clipping are rare). If the encoding error occurs randomly on each state, the effects can compensate each other. However, in NEQR, the data is encoded in a sparse space, or equivalently the substate is expressed by a one-hot vector; e.g., for the case of four qubit with 16 color levels, only one substate has nonzero amplitude and the other 15 substates have zero amplitude. Since the similarity score directly relates to the inner product of the noised one-hot vectors, it is susceptible to the encoding error.

-
- [1] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations* (Springer, Berlin, 1972), pp. 85–103.
- [2] N. Wiebe, A. Kapoor, and K. Svore, Quantum nearest-neighbor algorithms for machine learning, *Quantum Inf. Comput.* **15**, 318 (2015).
- [3] A. Gilliam, S. Woerner, and C. Goniculea, Grover adaptive search for constrained polynomial binary optimization, *Quantum* **5**, 428 (2021).
- [4] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
- [5] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (ACM Press, New York, 1996), pp. 212–219.
- [6] L. K. Grover, Synthesis of Quantum Superpositions by Quantum Computation, *Phys. Rev. Lett.* **85**, 1334 (2000).
- [7] Y. R. Sanders, G. H. Low, A. Scherer, and D. W. Berry, Black-Box Quantum State Preparation without Arithmetic, *Phys. Rev. Lett.* **122**, 020502 (2019).
- [8] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, *Phys. Rev. A* **83**, 032302 (2011).
- [9] V. V. Shende, S. S. Bullock, and I. L. Markov, Synthesis of quantum-logic circuits, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **25**, 1000 (2006).
- [10] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, *Phys. Rev. A* **78**, 052310 (2008).
- [11] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [12] J. Liu and H. Zhou, Hardware efficient quantum search algorithm, [arXiv:2103.14196](https://arxiv.org/abs/2103.14196).
- [13] V. Kasirajan, Quantum algorithms, in *Fundamentals of Quantum Computing: Theory and Practice*, edited by V. Kasirajan (Springer International Publishing, Cham, 2021), pp. 267–363.
- [14] K. Nakaji, S. Uno, Y. Suzuki, R. Raymond, T. Onodera, T. Tanaka, H. Tezuka, N. Mitsuda, and N. Yamamoto, Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicator, [arXiv:2103.13211](https://arxiv.org/abs/2103.13211).
- [15] M. Briński, J. Gwinner, V. Hlembotskyi, W. Jarnicki, S. Pliś, and A. Szady, Introducing structure to expedite quantum searching, *Phys. Rev. A* **103**, 062425 (2021).
- [16] J. Preskill, Quantum computing in the nisq era and beyond, *Quantum* **2**, 79 (2018).
- [17] Y. Ruan, X. Xue, and Y. Shen, Quantum image processing: Opportunities and challenges, *Math. Prob. Eng.* **2021**, 1 (2021).
- [18] N. Jiang, Y. Dang, and J. Wang, Quantum image matching, *Quantum Inf. Process.* **15**, 3543 (2016).
- [19] R.-G. Zhou, X. Liu, C. Zhu, L. Wei, X. Zhang, and H. Ian, Similarity analysis between quantum images, *Quantum Inf. Process.* **17**, 121 (2018).
- [20] X. Liu, R.-G. Zhou, A. El-Rafei, F.-X. Li, and R.-Q. Xu, Similarity assessment of quantum images, *Quantum Inf. Process.* **18**, 244 (2019).
- [21] X. Guanlei, X. Xiaogang, W. Xun, and W. Xiaotong, A novel quantum image parallel searching algorithm, *Optik* **209**, 164565 (2020).
- [22] A. M. Iliyasu, F. Yan, and K. Hirota, Metric for estimating congruity between quantum images, *Entropy* **18**, 360 (2016).
- [23] Y. Dang, N. Jiang, H. Hu, and W. Zhang, Analysis and improvement of the quantum image matching, *Quantum Inf. Process.* **16**, 269 (2017).
- [24] A. A. Ezhov, A. V. Nifanova, and D. Ventura, Quantum associative memory with distributed queries, *Inf. Sci.* **128**, 271 (2000).
- [25] K. Nakaji, Faster amplitude estimation, *Quantum Inf. Comput.* **20**, 1109 (2020).
- [26] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, *Quantum Inf. Process.* **19**, 75 (2020).
- [27] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).

- [28] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, *npj Quantum Inf.* **5**, 103 (2019).
- [29] N. Ahmed and K. R. Rao, Walsh-Hadamard transform, in *Orthogonal Transforms for Digital Signal Processing*, edited by N. Ahmed and K. R. Rao (Springer, Berlin, 1975), pp. 99–152.
- [30] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit born machines, *Phys. Rev. A* **98**, 062324 (2018).
- [31] B. Coyle, D. Mills, V. Danos, and E. Kashefi, The Born supremacy: Quantum advantage and training of an Ising Born machine, *npj Quantum Inf.* **6**, 60 (2020).
- [32] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Quantum circuits with uniformly controlled one-qubit gates, *Phys. Rev. A* **71**, 052330 (2005).
- [33] MD S. Anis, H. Abraham, AduOffei, R. Agarwal, G. Agliardi, M. Aharoni, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, S. Anagolum, E. Arbel, A. Asfaw, A. Athalye, A. Avkhadiev, C. Azaustre, P. Bhole, A. Banerjee, S. Banerjee, W. Bang, A. Bansal, P. Barkoutsos, A. Barnawal, G. Barron, G. S. Barron, L. Bello *et al.*, Qiskit: An open-source framework for quantum computing (2021), doi:10.5281/zenodo.2573505.
- [34] D. P. Kingma and L. J. Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA* (Ithaca, NY, 2015), p. 13.
- [35] D. Maslov, Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization, *Phys. Rev. A* **93**, 022311 (2016).
- [36] H. L. Tang, V. O. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor, *PRX Quantum* **2**, 020310 (2021).
- [37] N. V. Tkachenko, J. Sud, Y. Zhang, S. Tretiak, P. M. Anisimov, A. T. Arrasmith, P. J. Coles, L. Cincio, and P. A. Dub, Correlation-informed permutation of qubits for reducing ansatz depth in the variational quantum eigensolver, *PRX Quantum* **2**, 020337 (2021).
- [38] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [39] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, *Quantum* **3**, 214 (2019).
- [40] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nat. Commun.* **12**, 1791 (2021).
- [41] T. Volkoff and P. J. Coles, Large gradients via correlation in random parameterized quantum circuits, *Quantum Sci. Technol.* **6**, 025008 (2021).
- [42] L. K. Grover, Trade-offs in the quantum search algorithm, *Phys. Rev. A* **66**, 052314 (2002).
- [43] V. Hlembotskyi, R. Burczyński, W. Jarnicki, A. Szady, and J. Tułowiecki, Efficient unstructured search implementation on current ion-trap quantum processors, [arXiv:2010.03841](https://arxiv.org/abs/2010.03841).
- [44] J. Gwinner, M. Briański, W. Burkot, Ł. Czerwiński, and V. Hlembotskyi, Benchmarking 16-element quantum search algorithms on superconducting quantum processors, [arXiv:2007.06539](https://arxiv.org/abs/2007.06539).
- [45] K. Zhang, P. Rao, K. Yu, H. Lim, and V. Korepin, Implementation of efficient quantum search algorithms on NISQ computers, *Quantum Inf. Process.* **20**, 233 (2021).
- [46] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, *Fortschr. Phys.* **46**, 493 (1998).
- [47] Z. Wang, M. Xu, and Y. Zhang, Review of quantum image processing, *Arch. Comput. Methods Eng.* **29**, 737 (2022).
- [48] P. Q. Le, F. Dong, and K. Hirota, A flexible representation of quantum images for polynomial preparation, image compression, and processing operations, *Quantum Inf. Process.* **10**, 63 (2011).
- [49] X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, W. Zheng, J. Li, M. Zhao, X. Peng, and D. Suter, Quantum Image Processing and Its Application to Edge Detection: Theory and Experiment, *Phys. Rev. X* **7**, 031041 (2017).
- [50] Y. Zhang, K. Lu, Y. Gao, and M. Wang, NEQR: a novel enhanced quantum representation of digital images, *Quantum Inf. Process.* **12**, 2833 (2013).
- [51] UCI machine learning repository: Optical recognition of handwritten digits data set, <https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits>.