Multi-state Swap Test Algorithm

Wen Liu¹,²,³ & Han-Wen Yin² & Zhi-Rao Wang² & Wen-Qin Fan¹,²

1. State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, China
2. School of Computer Science and Cybersecurity, Communication University of China, Beijing, China
3. Key Laboratory of Convergent Media and Intelligent Technology (Communication University of China), Ministry of Education, Beijing, China

E-mail: lw8206@163.com

April 2022

Abstract. Estimating the overlap between two states is an important task with several applications in quantum information. However, the typical swap test circuit can only measure a sole pair of quantum states at a time. In this study we designed a recursive quantum circuit to measure overlaps of multiple quantum states |φ₁...φₙ⟩ concurrently with O(n log n) controlled-swap (CSWAP) gates and O(log n) ancillary qubits. This circuit enables us to get all pairwise overlaps among input quantum states |⟨φ₁|φ₂⟩|². Compared with existing schemes for measuring the overlap of multiple quantum states, our scheme provides higher precision and less consumption of ancillary qubits. In addition, we performed simulation experiments on IBM quantum cloud platform to verify the superiority of the scheme.

Keywords: Quantum information, The Swap Test, Multiple Quantum States, Overlap

Submitted to: J. Phys. A: Math. Gen.
1. Introduction

The concept of quantum computing was proposed by Richard Feynman in the early 1980s\[1\] and David Deutsch later gave a quantum algorithm solution to a toy issue\[2\]. In recent years, Quantum algorithms and their applications in fields like encryption\[3\], database search\[4\], quantum simulation\[5\], optimization problems\[6\] and linear systems of equations\[7\] (refer to \[8\] for more information) have proved spectacular. For certain problems, these quantum algorithms outperform classical algorithms in terms of speed. Typically, quantum algorithms are described using quantum circuits, which are made up of a sequence of quantum gates that can handle quantum bits. When dealing with quantum circuits, it’s important to measure the similarity or overlap between two states |ϕ⟩ and |ψ⟩, which can be denoted by |⟨ϕ,ψ⟩|²\[9\].

Swap Test is a well-known quantum circuit, which uses a controlled-swap (CSWAP) gate and an ancillary qubit to estimate the inner product |⟨ϕ,ψ⟩|² of two states |ϕ⟩, |ψ⟩\[10\][11]. It has attracted interest as a fundamental primitive and its efficient implementation on near-term quantum computers is a hot research issue at the moment. In\[12\], L. Cincio et al. presented a machine learning approach for discovering short-depth algorithm to compute states overlap, whose performance is better than the linear scaling of Swap Test. In \[13\], M. Fanizza et al. studied the estimation of overlap between two unknown pure quantum states in a finite-dimensional system, given M and N copies of each type. They demonstrated that allowing for general collective measurements on all copies can yield a more precise estimate. In \[14\], U. Chabaud et al. proposed a strategy for a programmable projective measurement device, which may be interpreted as an optimal Swap Test when only one copy of one state and M − 1 of the other is available. There are many applications that require state overlap computation thus making Swap Test appears as a subroutine in these applications. For instance, it is used to measure entanglement and indistinguishability for quantum states\[11\][15], and also common in quantum supervised learning to measure the distances of quantum states\[16\][17][18]. What’s more, proofs-of-concept in implementations of quantum computers\[19\][20] call for this Swap Test as well.

Consider the case when there are n quantum states |ϕ₁⟩, |ϕ₂⟩, ..., |ϕₙ⟩ and the Swap Test can only assess the overlap of two at a time. In order to obtain the overlap |⟨ϕᵢ|ϕⱼ⟩|² between arbitrary two states |ϕᵢ⟩, |ϕⱼ⟩ (i, j = 1, 2, ..., m; i ≠ j), there is a trivial solution which requires \(\frac{n(n-1)}{2}\) Swap Tests and \(\frac{n(n-1)}{2}\) ancillary qubits. In \[21\], X. Gitiaux et al. extended two-state Swap Test to n quantum states |φ₁⟩, |φ₂⟩, ..., |φₙ⟩. They built a unitary U₄ circuit with three ancillaries, three controlled-swap (CSWAP) gates, and one Swap Test to compute overlap for four states. Based on the U₄ circuit, they developed a recursive algorithm to generalize a circuit which can estimate overlaps between n quantum states at a time. The circuit requires \(O(n)\) CSWAP gates and \(O(\log n)\) ancillary qubits.

In this paper, following the idea of \[21\], we design a new circuit U₄ with two ancillaries,
Multi-state Swap Test algorithm

two controlled-swap (CSWAP) gates, and two simple Swap Test. We also present two rules to swap four groups of quantum states. Based on the swap rules and our $U_4$, a circuit to calculate the overlap between different pairs of $n$ arbitrary input states are presented. Our circuit requires $n \log_2 n$ CSWAP gates and $2(n - 1)$ ancillary qubits.

The paper is organized as follows: some correlative preliminaries are introduced in Section 2; a multi-state Swap Test algorithm is proposed in Section 3; In Section 4, the simulation of 8-state Swap Test algorithm on IBM Quantum Experience’s platform is described. And the correctness and performance of the algorithm are analyzed in section 5. A brief discussion and a summary are given in Section 6.

2. Preliminaries

2.1. Swap Test and Its Quantum Circuit

The Swap Test is used to estimate the overlap between two unknown quantum states, and it has a variety of applications. Given two unknown quantum states $|\phi\rangle$ and $|\psi\rangle$, the overlap between them is defined as $|\langle\phi|\psi\rangle|^2$, i.e. the inner product between their state vectors. The Swap Test quantum circuit for estimating the overlap between $|\phi\rangle$ and $|\psi\rangle$ is shown in Fig.(1):

![Figure 1. Quantum circuit implementing the Swap Test for two quantum states](image)

where the first qubit is an auxiliary qubit used for measurement and the other two qubits are used for input. According to the quantum circuit, the output denoted by $|\rho\rangle$ can be calculated as:

$$
|\rho\rangle = |0\rangle|\phi\rangle|\psi\rangle
\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\phi\rangle|\psi\rangle
\rightarrow \frac{1}{\sqrt{2}}(|0\rangle|\phi\rangle|\psi\rangle + |1\rangle|\psi\rangle|\phi\rangle)
\rightarrow \frac{1}{2}|0\rangle (|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle) + \frac{1}{2}|1\rangle (|\phi\rangle|\psi\rangle - |\psi\rangle|\phi\rangle)
$$

According to Equ.(1), the probability of measuring $|0\rangle$ is $Prob(0) = \frac{1+|\langle\phi|\psi\rangle|^2}{2}$.

Obviously, the overlap $\langle\phi|\psi\rangle$ is related to the probability of measuring $|0\rangle$. So we can derive:

$$
|\langle\phi|\psi\rangle|^2 = 2Prob(0) - 1
$$

In [15], Juan Carlos Garcia-Escartin et al. show that the Hong-Ou-Mandel effect from quantum optics is equivalent to the Swap Test. Then, they give a destructive Swap Test
module that can provide better performance than the traditional Swap Test. The basic implementation of their Swap Test costs three quantum registers and a CSWAP gate to estimate the overlap between two unknown quantum states. The CSWAP circuit is inspired by classical XOR swapping, and its quantum implementation is shown in Fig.(2).

\[ \text{Figure 2. Swap Test for qubit states with a CSWAP gate.} \]

X gate can get with the help of Z gate. So we have the equivalent quantum circuit of Swap Test in Fig.(3).

\[ \text{Figure 3. Swap Test circuit with a CCZ gate.} \]

Obviously, the ancillary qubit is not affected by the tested qubits after CCZ gate. We can just get rid of the last H and CNOT gates and perform measurements directly after the CCZ gate with no effect on the ancillary qubit and the result of the Swap Test. Moreover, with the help of a trick called ‘phase kickback’, we can rewrite the circuit as in Fig.(4).

\[ \text{Figure 4. Swap Test advancing the measurement} \]

The output of ancillary qubit can be predicted according to the output of tested qubits in the classical method, which is sometimes called the principle of deferred measurement. Clearly, we can just ignore the ancillary qubit and perform a Swap Test with the circuit in Fig.(5), which is the improved Swap Test.
2.2. Gitiaux’s Swap Test for Multiple Quantum States

In [19], Xavier Gitiaux et al. gave a new quantum algorithm that is used to estimate the overlap between multiple quantum states based on the standard two-state Swap Test and CSWAP gate.

First, they define a unitary $U_4$ for a 4-state circuit as a building block to develop a recursive algorithm to produce an $n$-state circuit.

As shown in Fig.(6), there are a total of four input states which can form 6 pairs of combinations. It requires at least 6 orthogonal states to label them. Therefore, the minimum number of ancillary qubits is three, and the minimum number of CSWAP gates is also three, each acting as control once.

Terms contained in the superposition state are listed in Table 1, where $s_1, s_2, s_3$ denote the basis states of the three ancillary qubits from bottom to top in the ancillary section in Fig.(6). The four state registers are labeled by $q_i (i = 1, 2, 3, 4)$. The input state is $|\phi_1\phi_2\phi_3\phi_4\rangle$, and the order of the output states depends on the measured values of the auxiliary qubits. As shown in the table 1, the output state will be $|\phi_4\phi_3\phi_2\phi_1\rangle$ if the ancillary qubits are $|011\rangle$. The essential of this algorithm is that each possible overlap between two states can be generated by only measuring the first two qubits, which can be verified by Table.(1).

And the $U_8$ for 8 inputs can be constructed from $U_4$ which is shown in Fig.(7). The
Multi-state Swap Test algorithm

| s1s2 | |00⟩ | |01⟩ | |10⟩ | |11⟩ |
|---|---|---|---|---|---|
| s3 = |0⟩ | |ϕ₁ϕ₂ϕ₃ϕ₄⟩ | |ϕ₁ϕ₄ϕ₃ϕ₂⟩ | |ϕ₂ϕ₃ϕ₁ϕ₄⟩ | |ϕ₄ϕ₂ϕ₁ϕ₃⟩ |
| s3 = |1⟩ | |ϕ₁ϕ₃ϕ₂ϕ₄⟩ | |ϕ₄ϕ₃ϕ₂ϕ₁⟩ | |ϕ₃ϕ₁ϕ₂ϕ₄⟩ | |ϕ₄ϕ₁ϕ₂ϕ₃⟩ |

Table 1. The swap results of $s_1, s_2, s_3$ and $q_1, q_2, q_3, q_4$

![Quantum circuit implementing $U_8$ for 8 quantum states](image)

Figure 7. Quantum circuit implementing $U_8$ for 8 quantum states

circuit is built into two sections of three ancillaries and three groups of three CSWAP gates, which amounts to six ancillaries and nine CSWAP gates. It is obvious that the quantum circuit is divided into two groups of four registers: $q_1q_2q_3q_4$ and $q_5q_6q_7q_8$ respectively. In the first step, the corresponding registers in the two sets are controlled by a set of three ancillaries in parallel which brings all six pairs in $|ϕ₁⟩, |ϕ₂⟩, |ϕ₃⟩, |ϕ₄⟩$ to $q_1q_2$ and all six pairs in $|ϕ₅⟩, |ϕ₆⟩, |ϕ₇⟩, |ϕ₈⟩$ to $q_5q_6$ respectively. Then another set of three ancillaries is introduced to control the pairing among $q_1q_2q_5q_6$ which makes the pairing between all 8 registers possible.

Next we consider the case $n = 2^k$. As shown in Fig.(8), we build the $U_n$ by repeating the same thing we did in $U_8$. Denoting the number of ancillaries in $U_n$ as $d_n$ and the number of CSWAP gate as $c_n$, We have $d_n = 3k - 3$ and $c_n = 3(2^{k-1} - 1)$ where $k = \log_2 n$.

![Quantum circuit implementing $U_n$ for $n$ quantum states](image)

Figure 8. Quantum circuit implementing $U_n$ for $n$ quantum states

To clarify, the procedure is to use $U_4$ within different groups first, and then on the
resultant registers of each group. $U_4$ acts each time to allow the pairing of two parts of the registers (these parts could be the simplest group or a group already combined by $U_4$ in the previous step.)

3. New Multi-state Swap Test algorithm

Suppose that there are $n$ quantum states $|\phi_1\rangle, |\phi_2\rangle, ..., |\phi_n\rangle$. An algorithm to construct one circuit to take all $n$ input states at once and get the overlap of arbitrary two input states in $|\phi_1\rangle, |\phi_2\rangle, ..., |\phi_n\rangle$ is presented.

Two swap rules are designed as follows: $N(N = 2^2, 2^3, ..., 2^k)$ quantum states can be divided into four groups $G_1, G_2, G_3, G_4$. Rule 1 is to exchange the states of 2nd and 3rd group in turn and get $G_1, G_3, G_2, G_4$; Rule 2 is to exchange the states of 2nd and 4th group in turn and get $G_1, G_4, G_3, G_2$. The multi-state Swap Test algorithm based on the two swap rules is as follows:

1. Constructing a circuit $U_4$ to estimate the overlap of any two of the 4 states quantum states $|\phi_1\rangle, |\phi_2\rangle, |\phi_3\rangle, |\phi_4\rangle$. In $U_4$, there are two ancillary qubits $s_1, s_2$ and four input qubits $q_1, q_2, q_3, q_4$. The initial states are $s_1 s_2 = |+\rangle |+\rangle$ and $q_1 q_2 q_3 q_4 = |\phi_1\rangle |\phi_2\rangle |\phi_3\rangle |\phi_4\rangle$. $q_1 q_2 q_3 q_4$ are divided into four groups $G_1 = q_1; G_2 = q_2; G_3 = q_3; G_4 = q_4$. There are also two CSWAP gates, where $s_1 (s_2)$ is the control qubit to swap two groups $G_2, G_3 (q_2, q_4)$ according to rule 1 (rule 2). Two simple Swap Test for two quantum states are places at the end of $U_4$ to measure the overlap for $q_1, q_2$ and $q_3, q_4$. The $U_4$ is shown in Fig.(9).

![Figure 9. Quantum circuit implementing $U_4$ for 4 quantum states](image)

The swap results between $s_1, s_2$ and $q_1, q_2, q_3, q_4$ are shown in Table.(2).

| $s_1 s_2$ | $|00\rangle$ | $|01\rangle$ | $|10\rangle$ | $|11\rangle$ |
|------------|---------------|---------------|---------------|---------------|
| $q_1 q_2 q_3 q_4$ | $|\phi_1\phi_2\phi_3\phi_4\rangle$ | $|\phi_1\phi_3\phi_2\phi_4\rangle$ | $|\phi_1\phi_4\phi_3\phi_2\rangle$ | $|\phi_1\phi_4\phi_2\phi_3\rangle$ |

Table 2. The swap results of $s_1, s_2$ and $q_1, q_2, q_3, q_4$

According to the Table.(2), if $s_1 s_2 = |00\rangle$, the overlap of $|\phi_1\phi_2\rangle$ and $|\phi_3\phi_4\rangle$ can be obtained through two simple Swap Test; if $s_1 s_2 = |01\rangle$, the overlap of $|\phi_1\phi_3\rangle$ and $|\phi_2\phi_4\rangle$...
can be obtained through two simple Swap Test; if \(s_1s_2 = |10\rangle\), the overlap of \(|\phi_1\phi_4\rangle\) and \(|\phi_3\phi_2\rangle\) can be obtained through two simple Swap Test.

(2) Constructing a circuit \(U_8\) to estimate the overlap of any two quantum states in \(|\phi_1\rangle, |\phi_2\rangle, |\phi_3\rangle, |\phi_4\rangle, |\phi_5\rangle, |\phi_6\rangle, |\phi_7\rangle, |\phi_8\rangle\). In \(U_8\), there are four ancillary qubits \(s_1, s_2, s_3, s_4\) and eight input qubits \(q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\). The initial states are \(s_1s_2s_3s_4 = |+\rangle |+\rangle |+\rangle |+\rangle\) and \(q_1q_2q_3q_4q_5q_6q_7q_8 = |\phi_1\phi_2\phi_3\phi_4\phi_5\phi_6\phi_7\phi_8\rangle\). \(q_1q_2q_3q_4q_5q_6q_7q_8\) are divided into four groups \(G_1 = q_1q_2; G_2 = q_3q_4; G_3 = q_5q_6; G_4 = q_7q_8\). There are also eight CSWAP gates. \(s_1(s_2)\) is a control qubit on two CSWAP gates to swap \(G_2, G_3(G_2, G_4)\) according to rule1(rule2). And \(s_3, q_1, q_2, q_3, q_4\) can be seen as inputs of \(U_4\) and \(s_4, q_5, q_6, q_7, q_8\) can also be seen as inputs of \(U_4\). Four simple Swap Test for two quantum states are places at the end of \(U_8\) to measure the overlap for \(q_1, q_2\) and \(q_3, q_4\). The \(U_8\) is shown in Fig.(10).

![Quantum circuit implementing \(U_8\) for 8 quantum states](image)

The swap results between \(s_1, s_2, s_3, s_4\) and \(q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\) are shown in Table.(3).

| \(s_1s_2s_3s_4\) | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|-------------------|------|------|------|------|------|------|------|------|
| \(q_1q_2q_3q_4q_5q_6q_7q_8\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) |

| \(s_1s_2s_3s_4\) | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|-------------------|------|------|------|------|------|------|------|------|
| \(q_1q_2q_3q_4q_5q_6q_7q_8\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) | \(|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle\) |

Table 3. The swap results of \(s_1, s_2, s_3, s_4\) and \(q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\)

(3) Constructing a circuit \(U_m\) to estimate any two quantum states’ overlap for \(m\) states \(|\phi_1\rangle, |\phi_2\rangle, \ldots, |\phi_m\rangle\). If \(2^{k-1} < m \leq 2^k, 2^k - m\) \(|0\rangle\) are added into \(m\) states. In \(U_n(n = 2^k)\), there are \(2(k - 1)\) ancillary qubits \(s_1, s_2, \ldots, s_2(k-1)\) and \(2^k\) input states \(q_1, q_2, \ldots, q_{2^k}\). The initial states are \(s_i = |+\rangle\) and \(q_i = |\phi_i\rangle\) \((i = 1, 2, \ldots, 2^k)\). \(q_1, q_2, \ldots, q_{2^k}\) are divided into four groups \(G_1 = q_1, q_2, \ldots, q_{2^{k-2}}, G_2 = q_{2^{k-2}+1}, q_{2^{k-2}+2}, \ldots, q_{2^{(k-2)}}\), \(G_3 = \ldots\).
For example, $q$ and the times of overlap results are close to the exact overlap results. The overlap of arbitrary two states $o$ by simple Swap Test for two quantum states are places at the end of $U$ for two quantum states are places at the end of $U$. After recursively constructing round by round, $s_{(2(k-1)-2^{k-1})}q_{4n-3},q_{4n-2},q_{4n-1},q_{4n}(i = 1, 2, ..., 2^{k-1})$ can be used as inputs of $U_4$. 2$k^{-1}$ simple Swap Test for two quantum states are places at the end of $U_{2k}$ to measure the overlap for $(q_1, q_2), (q_3, q_4),...,(q_{2k-1}, q_{2k})$.

4. Simulation of multi-state Swap Test algorithm

In this section, 8 random states $|\phi_i\rangle (i = 1, 2, ..., 8)$ are chosen as an example of a multi-state Swap Test algorithm and then the correctness of the algorithm on this example is verified by experiments on IBM quantum cloud platform.

For arbitrary two states $|\phi_i\rangle, |\phi_j\rangle (i, j = 1, 2, ..., 8; i \neq j)$, the exact overlap is denoted by $o(|\phi_i\rangle, |\phi_j\rangle)$, which can be calculated by the definition of overlap $|\langle \phi, \psi \rangle|^2$; And the estimated overlap is denoted by $\hat{o}(|\phi_i\rangle, |\phi_j\rangle)$, which can be obtained by executing simulated experiments.

The simulated quantum circuit on IBM quantum cloud platform for 8-state Swap Test algorithm is given in Fig.(11), where $q_0q_1q_2q_3$ are the auxiliary control qubits that corresponds to $s_1s_2s_3s_4$, $q_4q_5q_6q_7q_8q_9q_{10}q_{11}$ are the initial states that corresponds to $|\phi_1\rangle|\phi_2\rangle|\phi_3\rangle|\phi_4\rangle|\phi_5\rangle|\phi_6\rangle|\phi_7\rangle|\phi_8\rangle$ and $q_1q_2q_3q_4q_5$ are the result qubits of simple two-state Swap Test. The initial states $q_0q_1q_2q_3$ are $|+\rangle|+\rangle|+\rangle$. Pair of normalized random numbers are generated by Python random package and the initial states $q_4q_5q_6q_7q_8q_9q_{10}q_{11}$ in the experiment are $|\phi_1\rangle = [0.0864, 0.9963], |\phi_2\rangle = [0.8391, 0.5440], |\phi_3\rangle = [0.2202, 0.9755], |\phi_4\rangle = [0.5486, 0.8361], |\phi_5\rangle = [0.2607, 0.9654], |\phi_6\rangle = [0.4164, 0.9091], |\phi_7\rangle = [0.9981, 0.0609], |\phi_8\rangle = [0.4658, 0.8849]$. And the measurement results of this implementation through IBM Quantum Experience (QISKIT) are given in Fig.(12),with 8192 runs. Meanwhile, the measurement times of $q_0q_1q_2q_3q_1q_3q_1q_15$ is shown in Appendix A.

The estimated overlap of $|\phi_i\rangle, |\phi_j\rangle (i, j = 1, 2, ..., 8; i \neq j)$ can be obtained according to the relation of $s_1, s_2, s_3, s_4$ and $q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8$ in Table.2 and Appendix A. For example, $s_1, s_2, s_3, s_4 = |0010\rangle$ and $s_1, s_2, s_3, s_4 = |0011\rangle$, the measurement result of $q_{15}$ is related to estimate values of $\langle \phi_6 | \phi_7 \rangle$. After counting times of $q_{15} = |1\rangle$, $t_1 = 403$ and the times of $q_{15} = |0\rangle$, $t_0 = 601$, $\hat{o}(|\phi_6\rangle, |\phi_7\rangle) = \frac{2t_0}{t_0+t_1} - 1 = 0.4441$. The exact overlap of arbitrary two states $o(|\phi_i\rangle, |\phi_j\rangle)$ and the estimated overlap of arbitrary two states $\hat{o}(|\phi_i\rangle, |\phi_j\rangle)(i, j = 1, 2, ..., 8; i \neq j)$ is shown in Table.(4). The scatter diagram is shown in Fig.(13), where the estimate overlap value $\hat{o}(|\phi_i\rangle, |\phi_j\rangle)$ is X-axis and the exact overlap value $o(|\phi_i\rangle, |\phi_j\rangle)$ is Y-axis. It can be seen from Fig.(13) that most of the points fall near the diagonals of the picture, i.e. the line $y = x$, indicating that the estimate overlap results are close to the exact overlap results.
Figure 11. The circuit for 8-state Swap Test algorithm

Figure 12. Result of implementation of 8-state Swap Test algorithm
Multi-state Swap Test algorithm

The exact overlap value of arbitrary two states \(o(\ket{\phi_i}, \ket{\phi_j})\) and the estimate overlap value of arbitrary two states \(\tilde{o}(\ket{\phi_i}, \ket{\phi_j})\) (\(i, j = 1, 2, ..., 8; i \neq j\))

9 data sets of the initial states \(D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9\) are generated by Python random package to be simulated in this experiment and their initial values are shown in Appendix B. And the scatter diagrams of estimated overlaps and exact overlaps of these 9 sets are shown in Fig.(14), where the estimated overlaps is X-axis and the exact overlaps is Y-axis. It can be seen from Fig.6 that most of the points fall near the diagonals of the picture, i.e., the line \(y = x\), indicating that the estimated overlaps are close to the exact overlaps.

5. Analysis

In this section, we analyze the correctness and performance of our multi-state Swap Test algorithm in theory.

5.1. Correctness analysis

There is a transformation \(U_n\), which can take arbitrary unknown states \(\ket{\phi_1}, \ket{\phi_2}, ..., \ket{\phi_n} (n = 2^k (k > 1))\) as input states and \(2(k-1) \ket{+}\) as ancillaries. In \(U_n\), each pair \(\ket{\phi_{2i-1}}, \ket{\phi_{2i}} (i = 1, 2, ..., \frac{n}{2})\) can be labeled by ancillaries \(\ket{A_{(2i-1)2i}}\) by Lemma 4.1.
Lemma 4.1 Let $n = 2^k (k > 1)$, there exists a unitary $U_n$ that maps

$$U_n : |+\rangle^d_n \otimes_{i=1}^n |\phi_i\rangle \rightarrow |y_{ij}\rangle + |1_{ij}\rangle G_{ij}^{1'} \phi_i \phi_j G_{ij}^1 + ... + |m_{ij}\rangle G_{ij}^{m'} \phi_i \phi_j G_{ij}^m$$

where $|\phi_i\rangle, |\phi_j\rangle$ are arbitrary two input quantum states, and they are in adjacent quantum registers, i.e. in the quantum registers $(q_{2i-1}, q_{2i}), i \in \{1, 2, ..., \frac{n}{2}\}$. $|G_{ij}^{1'}\rangle, |G_{ij}\rangle, |y_{ij}\rangle$ are garbage states which is irrelevant to our target states, i.e. $|\phi_i\rangle, |\phi_j\rangle$. $|1_{ij}\rangle, ..., |m_{ij}\rangle$ represents the state in the first $d_n$ quantum registers. That means we can get all pairs $|\phi_i\rangle|\phi_j\rangle, i \neq j$ by measuring the specific quantum registers $(q_{2i-1}, q_{2i}), i \in \{1, 2, ..., \frac{n}{2}\}$ in the quantum circuit.

Constructive proof of Lemma 4.1 According to scientific induction, we assume that Lemma 4.1 is correct at first. Let the $n$ quantum registers divided into 4 groups $Q_1 = (q_1, ..., q_{n/4}), Q_2 = (q_{n/4+1}, ..., q_{n/2}), Q_3 = (q_{n/2+1}, ..., q_{3n/4}), Q_4 = (q_{3n/4+1}, ..., q_{n})$. Furthermore, we get 4 quantum states $|\Phi_1\rangle = |\phi_1...\phi_{n/4}\rangle, |\Phi_2\rangle = |\phi_{n/4+1}...\phi_{n/2}\rangle, |\Phi_3\rangle = |\phi_{n/2+1}...\phi_{3n/4}\rangle, |\Phi_4\rangle = |\phi_{3n/4+1}...\phi_n\rangle$. Applying $U_{n/2}$ to $(Q_1, Q_2)$ and $(Q_3, Q_4)$ respectively, we can get all pairs of input states $|\Phi_1\Phi_2\rangle$, i.e. $|\phi_1\rangle, ..., |\phi_{n/2}\rangle$, and the same
is true for the other $U_{n/2}$. If we apply a unitary $U_4$ on $(Q_1^1, Q_2^1, Q_3^1, Q_4^1)$ before the $U_{n/2}$, we will get all pairs of input states $|\phi_1\rangle, ..., |\phi_n\rangle$. For example, we want to get a pair $|\phi_a\phi_b\rangle, a \in [1, \frac{n}{4}], b \in [\frac{n}{2} + 1, \frac{3n}{4}]$. $|\Phi_1^1\Phi_3^3\rangle$ will be stored in $(Q_1^1, Q_2^1)$ if the ancillary qubits of $U_4$ is $|01\rangle$, and the pair $|\phi_a\phi_b\rangle$ might be measured after $U_{n/2}$.

Similarly, the correctness of $U_{n/2}$ can be provide by using $U_{n/4}$ and $U_4$, and so on. Finally, the correctness of $U_4$ is shown in Table.1. So Lemma 4.1 is proved. For the case that $n$ cannot be written in the form of $2^k$, we can add $2^k - n$ copy of any input state as the rest input.

So far, we have obtained all pairs of $n$ input quantum states, and can get each pair exactly according to the measurement results of auxiliary qubits. Therefore, we only need to apply the Swap Test in all specific registers $(q_1, q_2), ..., (q_{n-1}, q_n)$ to get the overlap between two input quantum states. As shown in Figure(16), the measurement results of $2(k - 1)$ auxiliaries determine the quantum state in each quantum register uniquely. Because it has been proved that all pairs of input quantum states have the opportunity to be measured in specific registers, all overlap estimates can be obtained after enough measurements.

### 5.2. Performance analysis

In this section, we compare our scheme and the scheme Swap Test for an arbitrary number of quantum states given by Gitiaux(referred to as SAN scheme later in this article).

#### 5.2.1. Precision

The precision of estimation is determined by the occurrences of each overlap with the total number of runs given, thus we can use this value, denoted by $m$, to represent the precision. For the SAN scheme, only one measurement can be obtained each time the quantum circuit is run. Therefore, in the case of $n = 2^k$ quantum state inputs, if the count of quantum circuit is run for $N$ times, each overlap is estimated to occur $\frac{2N}{n(n-1)}$ times in average, i.e., $m_1 = \frac{2N}{n(n-1)}$. For our scheme, one execution contains $\frac{n}{2}$ Swap Test operations. So in the same case, we have the precision $m_2 = \frac{N}{n-1}$, which means higher precision.

Fig.(15) shows the ratio of the average precision of our scheme with respect to SAN. Taking the number of the SAN scheme $m_1$ as the standard, it can be seen that with the increase of the circuits depth, the ratio between the new scheme and SAN scheme $\frac{m_2}{m_1}$ increases gradually, indicating that our precision advantage becomes gradually more pronounced as the circuit depth increases. The stepwise growth is due to the fact that we must apply such $k$ that $2^{k-1} < n \leq 2^k$. It should be noted that if the probability of occurrence of each quantum state is taken into account, the estimation of different overlaps will have different precision. This is because the number of recurrences of different pairs varies according to the table, which means that some pairs may occur
more frequently. However, this only depends on the artificial choices we make when coding, and the total effect does not have a large impact as the number increases.

5.2.2. Space complexity The space complexity of $U_n$ consists of the count of CSWAP gates and the number of ancillaries, where $n = 2^k$ and $k = \log_2 n$. First of all, we notice that the count of CSWAP gates $c_n$ follows the recursive relation:

$$c_n = 2c_{n/2} + \frac{n}{2}$$

where the first term comes from two $U_{n/2}$ and the second term comes from a $U_4$. The general term formula $c_n = nk$ can be obtained by simplifying the sequence relation. If considering the CSWAP gates come from the following Swap Test operation, we have $c_n = n(k + \frac{1}{2})$. Similarly, for SAN scheme, we can get the general term formula for the number of CSWAP gates $c'_n = 3(n - 1)$, which is shown in the right panel of Fig.(16). Furthermore, if we concentrate on the count of CSWAP gates, the complexity of $U_n$ in our scheme is $O(n \log n)$ and in SAN scheme is $O(n)$. Obviously, our scheme uses more CSWAP gates and has higher complexity than the SAN scheme.

As for the number of ancillaries, we have the relation:

$$d_n = d_{n-1} + 2$$

which means each time the number of input quantum states is doubled, 2 additional auxiliaries are required. It can be rewritten as $d_n = 2k - 2$, and the general term formula for SAN scheme the number of ancillaries is $d'_n = 3k - 3$. Obviously, our scheme uses fewer ancillaries than the SAN scheme, and both complexities are $O(\log n)$. The relationship between the number of auxiliaries and the number of input quantum states (assuming that all input quantum states are single quantum states) is shown in Figure(18). As shown in the left panel, the red line and the green line represent our scheme and SAN scheme respectively. It can be seen that the number of auxiliaries required by our new scheme is less than that of the SAN scheme.

If we consider the additional auxiliaries brought by the subsequent Swap Test of the
two Swap Test schemes, we can reduce it by using the Swap Test circuit optimization scheme mentioned in section 2 to reduce the auxiliaries through classical calculation. At this time, the total number of qubits required by our scheme will be less than that required by the SAN scheme, which means less resource consumption.

To sum up, our scheme provides a better estimation under the same count of circuit runs. At the same time, the number of auxiliaries required in our scheme is less than that of the SAN scheme. Although the complexity of the count of CSWAP gates in the SAN scheme is $O(n)$ which is less than that in our scheme $O(n \log n)$, When the circuits run, they are processed in parallel and do not affect the total time, so we trade a certain space cost for the desired time cost.

6. Summary

In this paper, we propose a new scheme for estimating overlap between all pair of input quantum states. Given arbitrary number of input quantum states $|\phi_1\rangle, ..., |\phi_n\rangle$, the scheme provide a way to calculate the overlap $|\langle \phi_i | \phi_j \rangle|^2$ for all pair of input quantum states where $i < j$, $i, j \in 1, 2, ..., n$.

Our scheme is based on the scheme given by [19]. The basic principle is we concentrate a unitary $U_n$ according to $n$ input quantum states $|\phi_1\rangle, ..., |\phi_n\rangle$, which is used to obtain a specific superposition state, so that we can get all pairs of input quantum states at a specific position ($q_{2n-1}, q_{2n}$), $n \in 1, 2, ..., \frac{n}{2}$. In order to decode the measurement results, we use $O(\log n)$ auxiliary qubits, their measurement results reveal what the quantum state in the register is. Finally, the results are obtained by performing Swap Test on these positions to estimate the value of overlap between each pair. A total of $O(n \log n)$ CSWAP gates are used in the quantum circuit. We carry out experiments on IBM cloud platform, and give a exact analysis to verify the correctness of our scheme. At the same time, we analyze the complexity and accuracy of our scheme. Compared with the
original scheme [19], our scheme uses fewer auxiliary qubits and has higher accuracy, but the line depth (the number of CSWAP gates) will be higher than the original scheme. Therefore, if it is possible to find a way to reduce the use of cswap gates, the performance of the scheme will be further improved.

Acknowledgments

This work was supported in part by the 2019 National Social Science Foundation Art Major Project, Network Culture Security Research, under Grant 19zd12, in part by the High-Quality and Cutting-Edge Disciplines Construction Project for Universities in Beijing (Internet Information, Communication University of China), and in part by the Fundamental Research Funds for the Central Universities, and by Equipment Pre-research Field Fund under Grant 61403110320.

References

[1] Shor P W 1982 *International Journal of exact Physics* 21 467-488
[2] Deutsch D 1985 *Proceedings of the Royal Society A Mathematical* 400 97-117
[3] Shor, Peter W 1999 *SIAM Review* 26 1484-1509
[4] Grover L K 1996 *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (STOC96)* 212-219
[5] Lloyd S 1996 *Science* 273 1073-1078
[6] Farhi E, Goldstone J, Gutmann S, et al, 2000 *Physics*
[7] Harrow A W, Hassidim A, Lloyd S 2009 *Physical Review Letters* 103 150502
[8] Jordan S (https://quantumalgorithmzoo.org/ (Accessed: 2020-11-07))
[9] Aimeur E, Zekrifa D M S, Gambs S 2006 *Proceedings of the 19th international conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence* (Springer-Verlag) 431-42
[10] Chuang I, Gottesman D 2001 *SIAM Review* (arXiv:quant-ph/0105032)
[11] Garcia-Escartin J C, Chamorro-Posada P 2013 *Physical Review A* 87 239-257
[12] Cincio L, Y Subaşı, Sornborger A T, et al, 2018 *New Journal of Physics* 20 113022
[13] Farizza M, Rosati M, Skotiniotis M, et al, 2020 *Physical review letters* 124 060503
[14] Chabaud U, Diamanti E, Markham D, et al, 2018 *Physical Review A* 98 062318
[15] Linke N M, Johri S, Figgatt C, et al, 2017 *APS March Meeting* (American Physical Society)
[16] Lloyd S, Mohseni M, Rebentrost P 2013 (arXiv:1307.0411)
[17] Wiebe N, Kapoor A, Svore K 2014 *Quantum Information & Computation* 15 318-358
[18] V Havlíček, AD Córcoles, Temme K, et al, 2019 *Nature* 567 209-212
[19] Schuld Maria, Sinayskiy Ilya, et al, 2015 (arXiv:1409.3097)
[20] Kang Min-Sung, Heo Jino, et al 2019 *Scientific Reports* 9 (1) 6167
[21] Gitiaux X, Morris I, Emelianenko M, et al, 2021 (arXiv:2110.13261)
Appendix

A

| $|q_0,q_2,q_4,q_6,q_8,q_{10},q_{12},q_{14},q_{15}\rangle$ | counts | $|q_0,q_2,q_4,q_6,q_8,q_{10},q_{12},q_{14},q_{15}\rangle$ | counts | $|q_0,q_2,q_4,q_6,q_8,q_{10},q_{12},q_{14},q_{15}\rangle$ | counts |
|---|---|---|---|---|
| $|01\rangle$ | 38 | $|01\rangle$ | 38 | $|01\rangle$ | 38 |
| $|01\rangle$ | 235 | $|01\rangle$ | 207 | $|01\rangle$ | 7 |
| $|10\rangle$ | 245 | $|10\rangle$ | 30 | $|10\rangle$ | 1 |
| $|00\rangle$ | 60 | $|10\rangle$ | 3 | $|00\rangle$ | 1 |
| $|01\rangle$ | 83 | $|10\rangle$ | 55 | $|01\rangle$ | 4 |
| $|00\rangle$ | 221 | $|11\rangle$ | 31 | $|00\rangle$ | 22 |
| $|10\rangle$ | 36 | $|10\rangle$ | 148 | $|10\rangle$ | 9 |
| $|11\rangle$ | 10 | $|10\rangle$ | 34 | $|11\rangle$ | 4 |
| $|11\rangle$ | 389 | $|10\rangle$ | 9 | $|10\rangle$ | 29 |
| $|01\rangle$ | 256 | $|10\rangle$ | 32 | $|01\rangle$ | 6 |
| $|10\rangle$ | 388 | $|01\rangle$ | 4 | $|01\rangle$ | 5 |
| $|10\rangle$ | 116 | $|11\rangle$ | 19 | $|11\rangle$ | 1 |
| $|01\rangle$ | 45 | $|11\rangle$ | 3 | $|01\rangle$ | 19 |
| $|01\rangle$ | 114 | $|10\rangle$ | 4 | $|01\rangle$ | 18 |
| $|11\rangle$ | 238 | $|00\rangle$ | 18 | $|00\rangle$ | 6 |
| $|11\rangle$ | 29 | $|00\rangle$ | 62 | $|00\rangle$ | 2 |
| $|00\rangle$ | 220 | $|01\rangle$ | 35 | $|01\rangle$ | 4 |
| $|01\rangle$ | 73 | $|11\rangle$ | 16 | $|11\rangle$ | 3 |
| $|10\rangle$ | 43 | $|11\rangle$ | 20 | $|11\rangle$ | 10 |
| $|00\rangle$ | 26 | $|10\rangle$ | 57 | $|10\rangle$ | 1 |
| $|11\rangle$ | 44 | $|10\rangle$ | 49 | $|10\rangle$ | 9 |
| $|01\rangle$ | 189 | $|00\rangle$ | 4 | $|00\rangle$ | 4 |
| $|11\rangle$ | 4 | $|10\rangle$ | 88 | $|10\rangle$ | 8 |
| $|11\rangle$ | 92 | $|10\rangle$ | 19 | $|10\rangle$ | 2 |
| $|11\rangle$ | 200 | $|01\rangle$ | 33 | $|01\rangle$ | 1 |
| $|10\rangle$ | 1 | $|11\rangle$ | 4 | $|11\rangle$ | 6 |
| $|00\rangle$ | 238 | $|10\rangle$ | 18 | $|10\rangle$ | 6 |
| $|00\rangle$ | 131 | $|10\rangle$ | 30 | $|10\rangle$ | 4 |
| $|11\rangle$ | 218 | $|01\rangle$ | 5 | $|01\rangle$ | 3 |
| $|10\rangle$ | 93 | $|01\rangle$ | 18 | $|01\rangle$ | 1 |
| $|00\rangle$ | 189 | $|11\rangle$ | 2 | $|11\rangle$ | 1 |
| $|11\rangle$ | 54 | $|11\rangle$ | 69 | $|11\rangle$ | 1 |
| $|00\rangle$ | 126 | $|01\rangle$ | 2 | $|01\rangle$ | 3 |
| $|11\rangle$ | 121 | $|00\rangle$ | 2 | $|00\rangle$ | 6 |
| $|01\rangle$ | 195 | $|11\rangle$ | 12 | $|11\rangle$ | 1 |
| $|00\rangle$ | 99 | $|10\rangle$ | 2 | $|10\rangle$ | 2 |
| $|11\rangle$ | 413 | $|00\rangle$ | 18 | $|00\rangle$ | 4 |
| $|11\rangle$ | 104 | $|10\rangle$ | 26 | $|10\rangle$ | 1 |
| $|01\rangle$ | 109 | $|01\rangle$ | 6 | $|01\rangle$ | 1 |
| $|00\rangle$ | 49 | $|11\rangle$ | 7 | $|11\rangle$ | 2 |
| $|10\rangle$ | 231 | $|01\rangle$ | 7 | $|01\rangle$ | 1 |
| $|10\rangle$ | 234 | $|11\rangle$ | 5 | $|11\rangle$ | 1 |
| $|01\rangle$ | 252 | $|10\rangle$ | 8 | $|10\rangle$ | 1 |
| $|00\rangle$ | 56 | $|00\rangle$ | 3 | $|00\rangle$ | 4 |
| $|11\rangle$ | 13 | $|11\rangle$ | 2 | $|11\rangle$ | 1 |
| $|10\rangle$ | 3 | $|10\rangle$ | 2 | $|10\rangle$ | 1 |
| $|10\rangle$ | 1 | $|00\rangle$ | 4 | $|00\rangle$ | 1 |
| $|00\rangle$ | 1 | $|01\rangle$ | 1 | $|01\rangle$ | 2 |
| $|11\rangle$ | 2 | $|01\rangle$ | 1 | $|01\rangle$ | 1 |
| $|11\rangle$ | 1 | $|11\rangle$ | 3 | $|11\rangle$ | 1 |
| $|10\rangle$ | 1 | $|11\rangle$ | 2 | $|11\rangle$ | 1 |
| $|10\rangle$ | 1 | $|00\rangle$ | 1 | $|00\rangle$ | 1 |

**Table 5.** The counts of different measurement result of ancillary qubits $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}$.
**Multi-state Swap Test algorithm**

\[ B \]

| \(D_1\) | \(D_2\) | \(D_3\) |
|--------|--------|--------|
| \(|\phi_1\rangle = [0.1340, 0.9910]\) | \(|\phi_3\rangle = [0.9697, 0.2443]\) | \(|\phi_1\rangle = [0.8240, 0.5665]\) |
| \(|\phi_2\rangle = [0.5666, 0.8240]\) | \(|\phi_2\rangle = [0.8136, 0.5814]\) | \(|\phi_2\rangle = [0.0398, 0.9992]\) |
| \(|\phi_3\rangle = [0.1262, 0.9920]\) | \(|\phi_3\rangle = [0.2681, 0.9634]\) | \(|\phi_3\rangle = [0.3343, 0.9425]\) |
| \(|\phi_4\rangle = [0.2899, 0.9571]\) | \(|\phi_4\rangle = [0.1963, 0.9806]\) | \(|\phi_4\rangle = [0.4898, 0.8719]\) |
| \(|\phi_5\rangle = [0.9746, 0.2238]\) | \(|\phi_5\rangle = [0.5862, 0.8102]\) | \(|\phi_5\rangle = [0.8140, 0.5808]\) |
| \(|\phi_6\rangle = [0.4602, 0.8878]\) | \(|\phi_6\rangle = [0.2324, 0.9726]\) | \(|\phi_6\rangle = [0.7047, 0.7095]\) |
| \(|\phi_7\rangle = [0.9813, 0.1926]\) | \(|\phi_7\rangle = [0.9565, 0.2917]\) | \(|\phi_7\rangle = [0.3366, 0.9416]\) |
| \(|\phi_8\rangle = [0.9695, 0.2452]\) | \(|\phi_8\rangle = [0.5461, 0.8377]\) | \(|\phi_8\rangle = [0.6131, 0.7900]\) |

| \(D_4\) | \(D_5\) | \(D_6\) |
|--------|--------|--------|
| \(|\phi_1\rangle = [0.2945, 0.9557]\) | \(|\phi_1\rangle = [0.4298, 0.9029]\) | \(|\phi_1\rangle = [0.0089, 1.0000]\) |
| \(|\phi_2\rangle = [0.6896, 0.7242]\) | \(|\phi_2\rangle = [0.5133, 0.8582]\) | \(|\phi_2\rangle = [0.1810, 0.9835]\) |
| \(|\phi_3\rangle = [0.1983, 0.9801]\) | \(|\phi_3\rangle = [0.3321, 0.9432]\) | \(|\phi_3\rangle = [0.6282, 0.7780]\) |
| \(|\phi_4\rangle = [0.3120, 0.9501]\) | \(|\phi_4\rangle = [0.9703, 0.2419]\) | \(|\phi_4\rangle = [0.6501, 0.7599]\) |
| \(|\phi_5\rangle = [0.6402, 0.7682]\) | \(|\phi_5\rangle = [0.1018, 0.9948]\) | \(|\phi_5\rangle = [0.4055, 0.9141]\) |
| \(|\phi_6\rangle = [0.0190, 0.9998]\) | \(|\phi_6\rangle = [0.1415, 0.9899]\) | \(|\phi_6\rangle = [0.4306, 0.9026]\) |
| \(|\phi_7\rangle = [0.6453, 0.7639]\) | \(|\phi_7\rangle = [0.3026, 0.9531]\) | \(|\phi_7\rangle = [0.1430, 0.9897]\) |
| \(|\phi_8\rangle = [0.7724, 0.6351]\) | \(|\phi_8\rangle = [0.1908, 0.9816]\) | \(|\phi_8\rangle = [0.2031, 0.9792]\) |

| \(D_7\) | \(D_8\) | \(D_9\) |
|--------|--------|--------|
| \(|\phi_1\rangle = [0.5632, 0.8263]\) | \(|\phi_1\rangle = [0.8502, 0.5265]\) | \(|\phi_1\rangle = [0.7207, 0.6932]\) |
| \(|\phi_2\rangle = [0.1064, 0.9943]\) | \(|\phi_2\rangle = [0.1541, 0.9881]\) | \(|\phi_2\rangle = [0.0995, 0.9950]\) |
| \(|\phi_3\rangle = [0.9086, 0.4176]\) | \(|\phi_3\rangle = [0.3624, 0.9320]\) | \(|\phi_3\rangle = [0.1926, 0.9813]\) |
| \(|\phi_4\rangle = [0.0576, 0.9983]\) | \(|\phi_4\rangle = [0.3696, 0.9292]\) | \(|\phi_4\rangle = [0.7677, 0.6408]\) |
| \(|\phi_5\rangle = [0.3325, 0.9431]\) | \(|\phi_5\rangle = [0.8279, 0.5608]\) | \(|\phi_5\rangle = [0.2066, 0.9784]\) |
| \(|\phi_6\rangle = [0.0025, 1.0000]\) | \(|\phi_6\rangle = [0.1571, 0.9876]\) | \(|\phi_6\rangle = [0.0124, 0.9999]\) |
| \(|\phi_7\rangle = [0.0900, 0.9960]\) | \(|\phi_7\rangle = [0.4334, 0.9012]\) | \(|\phi_7\rangle = [0.2994, 0.9541]\) |
| \(|\phi_8\rangle = [0.7227, 0.6912]\) | \(|\phi_8\rangle = [0.7817, 0.6237]\) | \(|\phi_8\rangle = [0.84579, 0.5335]\) |

**Table 6.** The initial states for \(D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9\)