Automatic Test Pattern Generation for Robust Quantum Circuit Testing

KEAN CHEN, Institute of Software, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China, and Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA
MINGSHENG YING, Centre for Quantum Software and Information, University of Technology Sydney, Australia

Quantum circuit testing is essential for detecting potential faults in realistic quantum devices, while the testing process itself also suffers from the inexactness and unreliability of quantum operations. This paper alleviates the issue by proposing a novel framework of automatic test pattern generation (ATPG) for robust testing of logical quantum circuits. We introduce the stabilizer projector decomposition (SPD) for representing the quantum test pattern, and construct the test application (i.e., state preparation and measurement) using Clifford-only circuits, which are rather robust and efficient as evidenced in the fault-tolerant quantum computation. However, it is generally hard to generate SPDs due to the exponentially growing number of the stabilizer projectors. To circumvent this difficulty, we develop an SPD generation algorithm, as well as several acceleration techniques which can exploit both locality and sparsity in generating SPDs. The effectiveness of our algorithms are validated by 1) theoretical guarantees under reasonable conditions, 2) experimental results on commonly used benchmark circuits, such as Quantum Fourier Transform (QFT), Quantum Volume (QV) and Bernstein-Vazirani (BV) in IBM Qiskit.

CCS Concepts: • **Hardware → Test-pattern generation and fault simulation**; **Quantum computation**.

Additional Key Words and Phrases: Quantum circuit, circuit testing, ATPG

1 INTRODUCTION

Most quantum algorithms, such as Shor's algorithm [40], Grover's algorithm [18] and HHL algorithm [22], require execution on logical quantum circuits. While the quantum error-correcting codes provide some degree of fault tolerance for logical quantum circuits, the overall error rate remains non-negligible, especially at large scales. Therefore, developing effective testing methods for logical quantum circuits is essential for advancing large-scale quantum computing.

1.1 Challenges in Quantum ATPG

For classical circuit testing, an automatic test pattern generation (ATPG) algorithm called the D-algorithm, was first proposed by Roth [39], where a new logical value D is introduced to represent both the good and faulty circuit values. After that, many ATPG algorithms have been proposed and employed in industry. However, it is hard to directly adapt classical ATPG algorithms for quantum circuits. There are several challenges that impede quantum ATPG algorithms:

(1) **Computational complexity** — computation time required to generate test patterns. At first glance, one may assume that the reversibility of quantum circuits makes the fault-excitation and fault-propagation always possible and straightforward. After all, the reversibility property ensures that fault signals can traverse the circuit bidirectionally, reaching both the primary input and output of the circuit. However, this approach requires matrix multiplication with sizes exponential in the number of qubits, leading to significant inefficiency in computational complexity.
(2) **Test time complexity** — number of experiments required to obtain valid and reliable testing results. The probabilistic nature and inherent uncertainty of quantum mechanisms pose challenges in efficiently extracting information from a given quantum device. For instance, fault effects may manifest as small rotations in specific

Authors' addresses: Kean Chen, Institute of Software, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China, and Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA, keanchen.gan@gmail.com; Mingsheng Ying, Centre for Quantum Software and Information, University of Technology Sydney, Australia, Mingsheng.Ying@uts.edu.au.

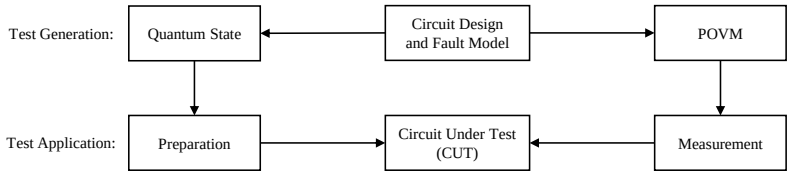


Fig. 1. Components of quantum ATPG.

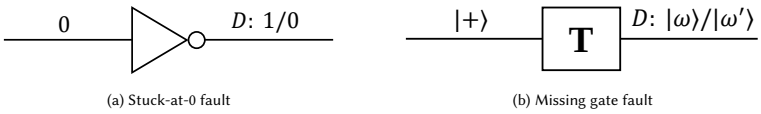
directions or subtle perturbations in the phase, resulting in low distinguishability between faulty and fault-free gates. Detecting such faults requires conducting numerous repeated experiments to obtain valid results with a sufficiently high probability (see Section 2.3 and Example 4 for more details). Consequently, the associated test time complexity can become prohibitive.

(3) **Test equipment complexity** — complexity of the quantum circuit required to implement the test pattern (i.e., complexity of the quantum equipment for test application, including state preparation and measurement, see Fig. 1). Examining the impact of test equipment complexity is crucial. Consider a straightforward scenario: suppose the fault-site resides at the end of a circuit, and the state capable of exciting the fault is simply $|0\rangle$. In this case, the primary input state should be $U^{-1}|0\rangle$, ensuring that the state-preparation circuit can trivially be U^{-1}, where U denotes the unitary corresponding to the Circuit Under Test (CUT). Interestingly, both U^{-1} and U share the same circuit complexity (up to a constant scalar). This seemingly innocuous detail has far-reaching implications. The complexity of the test equipment can be remarkably high, potentially rivaling that of the CUT itself. Furthermore, due to this elevated complexity, the test equipment is susceptible to non-negligible errors — often referred to as State Preparation and Measurement (SPAM) errors [20]. These errors render the test equipment itself faulty. Consequently, the complexity of the test equipment not only impacts efficiency but also significantly influences the robustness and reliability of the testing process.

1.2 Related Works

These challenges have been partially addressed or alleviated in several seminal works on quantum ATPG. Paler et al. [34] proposed the Binary Tomographic Test (BTT) which meets the probabilistic nature of quantum circuits. A BTT is a pair of test input state and test measurement. They first simulate both fault-free and faulty quantum circuits on the BTTs. Then the CUT is executed multiple times with the same BTTs. By analyzing the output distributions and comparing them to the simulation results, faults can be detected. However, in [34], the choice of test input states and test measurements is limited to the computational basis. While this approach simplifies test equipment complexity, it suffers from inefficiencies in test time complexity. Bera [5] improved the BTT algorithm to account for the variety of quantum fault models. He adopts the unitary discrimination protocol [3] with Helstrom measurement [24], where the input state and measurement are adaptively selected based on the specific fault being targeted. As a result, an arbitrary single fault can be detected with high probability, significantly improving test time complexity. However, it's worth noting that Bera's approach [5] sacrifices test equipment complexity, making it less robust in the context of quantum circuit testing.

On the other hand, Randomized Benchmarking (RB) [12, 30], a widely used technique in quantum information community, has achieved great successes in assessing quantum gate. One crucial advantage of randomized benchmarking



(a) Stuck-at-0 fault (b) Missing gate fault

Fig. 2. An example of the quantum analogue of classical D-value, where $|\omega\rangle/|\omega'\rangle$ corresponds to the Helstrom measurement for distinguishing quantum states $T|+\rangle$ and $|+\rangle$

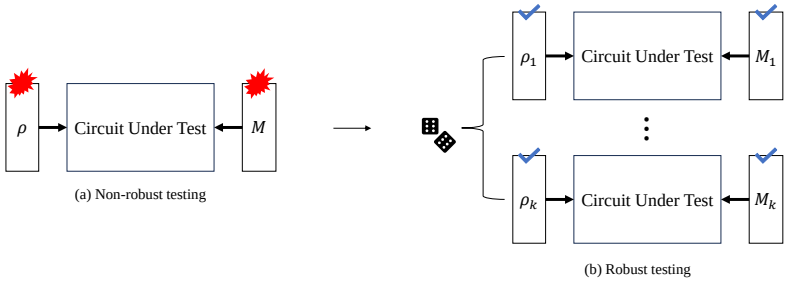
lies in its robustness against State Preparation and Measurement (SPAM) errors. However, RB relies on the gate-independent error assumption (or its weaker variant). Unfortunately, this assumption does not hold universally for logical (fault-tolerant) quantum circuits, since the implementations, and thus the fault models, of different logical gates vary greatly [13, 44]. Also, RB requires the ability to dynamically apply different gate sequences. This approach is well-suited for certain physical implementations, such as superconducting quantum circuits, where gates are dynamic pulses sequentially applied to reused physical qubits. However, photonic devices operate differently. Their gates are fixed hardware components integrated onto a chip [36, 41], resembling classical circuits. Due to this fixed gate structure, the assumption of a dynamic gate sequence may not hold for photonic devices. Moreover, faults can arise due to the integration of specific gates within a quantum circuit. These faults may not be solely dependent on an individual gate but can emerge when the gate is integrated into a circuit and interacts with other gates [20, 21]. In such cases, RB is not sufficient because it does not preserve or consider the overall circuit structure that we are concerned with. Therefore, unlike RB, which primarily focuses on testing individual quantum gates, there is a need for effective testing of an entire logical quantum circuit as a whole.

1.3 Main Idea

In this paper, we propose a novel framework of quantum ATPG. We begin by considering an quantum analogue of the classical D-algorithm. Suppose at the fault-site, say quantum gate U_i, we have a fault model U_i'. By the unitary discrimination protocol [3] and Helstrom measurement [24], we can obtain the input state $|\psi\rangle$ and projective measurement $\{|\omega\rangle\langle\omega|, |\omega'\rangle\langle\omega'|\}$ for optimal distinguishability between U_i and U_i' (i.e., ensuring optimal test time complexity). This measurement is analogue to the D-value in the classical D-algorithm. Indeed, it can be seen as a composite "logical value" $|\omega\rangle/|\omega'\rangle$ where the first entry $|\omega\rangle$ represents the "value" of fault-free gate and the second entry $|\omega'\rangle$ represents the "value" of faulty gate. A simple example is shown in Fig. 2. Then, similar to the D-algorithm, we propagate $|\psi\rangle$ and $|\omega\rangle/|\omega'\rangle$ to the primary input and output of the circuit to obtain the test pattern.

However, as previously mentioned, a critical challenge lies in the test equipment complexity and the necessity for robustness in test application (see the above discussion on challenge (3)). To address this issue, we decompose the high-complexity test pattern into a set of simpler test patterns —— each simpler test pattern is implemented via Clifford-only circuits, and these test patterns are combined through classical randomness, as shown in Fig. 3. For example, the direct implementation of the test patterns on a QFT_10 circuit leads to an average circuit size (i.e., the test equipment complexity) of 247, while our method leads to an average circuit size of only 29 (more details can be found in Table 3). Moreover, our method only uses Clifford circuits, which exhibit significant advantages in terms of both efficiency and robustness, as shown below:

(i) The Clifford gates are nearly ideal [7] in the logical (fault-tolerant) quantum circuits based on stabilizer codes, since the Clifford gates have rather robust and cheap implementation [14, 44].



(a) Non-robust testing

(b) Robust testing

Fig. 3. (a) Non-robust testing: the test pattern (ρ, M) is directly implemented, which is susceptible to SPAM errors (see the challenge (3)). (b) Robust testing: the test pattern is decomposed into simpler ones (i.e., (ρ_i, M_i)), which are implemented using Clifford-only circuits, and are then combined through classical randomness.

(ii) Any Clifford operation can be efficiently synthesized to a relatively small Clifford circuit with size no more than $O(n^2/\log n)$ [2, 8], where n is the number of qubits.

(iii) The Clifford-universal gate set (*e.g.* Hadamard, Phase and controlled-NOT) is finite and discrete. Thus the Clifford gates can be carefully calibrated on specific quantum devices. This contrasts with the continuous rotation gates such as those in the QFT circuit.

The above advantages ensure that using the Clifford circuits for test application is simpler and more robust against SPAM error than using general quantum circuits. It should be noted that our robust testing method introduces an additional overhead on the number of experiments. This can be viewed as a trade off between quantum resources and classical resources. Specifically, we transform the cost of quantum resources (i.e., complexity of quantum test equipment) to the cost of classical resources (i.e., repeated experiments sampled from classical randomness). This trade off is valuable in quantum circuit testing as the classical resources are more available and reliable for the near-term devices.

Based on this observation, we introduce the stabilizer projector decomposition (SPD) to represent the test pattern. By applying a sampling algorithm on the SPD, the test application can be constructed using Clifford-only circuits with classical randomness. The number of required experiments in our sampling algorithm is explicitly related to the 1-norm metric of SPD. The optimal SPD (*i.e.*, that with minimal 1-norm) for given test pattern can be obtained by solving a convex optimization problem. But in general, it is practically intractable to solve this optimization problem due to the exponentially growing of the problem size. To mitigate this issue, we propose an SPD generation algorithm with several acceleration techniques, where both locality and sparsity are exploited in the SPD calculation. It is proved that the locality exploiting algorithm is optimal in the sense that it reveals the most locality of given SPD, under some mild conditions. To demonstrate the effectiveness and efficiency of our approach, the overall quantum ATPG framework and the proposed SPD generation algorithm are implemented and evaluated on several commonly used benchmark circuits in IBM Qiskit. The code is available at Github[1].

1.4 Organization

Our quantum ATPG framework is visualized in Fig. 4. The paper is then organized as follows. In Section 1 (i.e. this section), we discussed 3 main challenges in quantum circuit testing, i.e., computational complexity, test time complexity

[1] https://github.com/cccorn/Q-ATPG

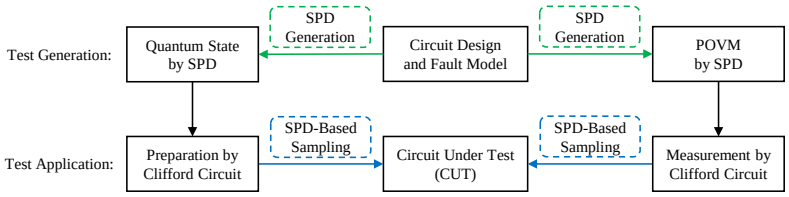


Fig. 4. Our quantum ATPG framework. Linked to Fig. 1, the quantum state and POVM are represented by SPDs, and the state preparation and measurement are implemented using Clifford circuits. The test application and test generation are conducted by the SPD-based sampling algorithm (see Section 3) and the SPD generation algorithm (see Section 4), respectively.

and test equipment complexity, followed by our main idea to handle these challenges. In Section 2, we provide the necessary background knowledge, and further explain the challenges with a series of detailed examples, i.e., Example 1 – 5. In Section 3, we present an SPD-based sampling algorithm (cf. the blue dashed box in Fig. 4), targeting on test equipment complexity. In Section 4, we present an efficient SPD generation algorithm (cf. the green dashed box in Fig. 4), targeting on both test time complexity and computational complexity. In Section 5, we experimentally evaluate the proposed method on different benchmark circuits.

2 PRELIMINARIES

2.1 Quantum Circuits

Quantum circuits are made up of qubits (quantum bits), quantum gates and measurements. Using the Dirac notation, a (pure) state of a single qubit is represented by $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ with complex numbers α_0 and α_1 satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$. More generally, a state of n qubits can be written as a superposition of n-bit strings:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle \tag{1}$$

where complex numbers α_x satisfies the normalization condition $\sum_x |\alpha_x|^2 = 1$. If we identify a string $x = x_1 \ldots x_n \in \{0,1\}^n$ with the integer $x = \sum_{i=1}^n x_i \cdot 2^{i-1}$, then this state can also be represented by the 2^n-dimensional column vector $|\psi\rangle = (\alpha_0, \ldots, \alpha_{2^n-1})^T$. In addition, we use $\langle\psi|$ to denote the row vector $(\alpha_0^*, \ldots, \alpha_{2^n-1}^*)$, where the symbol $*$ represents the complex conjugate. Given an ensemble of pure states $\{(p_i, |\psi_i\rangle)\}$ where $\sum_i p_i = 1$, and suppose that the quantum system is in one of the $|\psi_i\rangle$ with probability p_i. We then refer to this system as being in a mixed quantum state. Such a state can be conveniently represented using a mathematical tool called the density operator $\rho = \sum_i p_i|\psi_i\rangle\langle\psi_i|$. When the context is clear, the term "quantum state" will be used to mean either a pure state or a mixed state.

A quantum gate on n qubits is mathematically described by a $2^n \times 2^n$ unitary matrix $U = \{u_{ij}\}$ satisfying $U^\dagger U = I_n$, where $U^\dagger$ stand for the conjugate transpose of U and I_n the identity matrix of dimension 2^n (we may omit the subscript of I_n when it does not cause confusion). If a pure state $|\psi\rangle$ (or mixed state ρ) is input to this gate, then the output from the gate is represented by the pure state $U|\psi\rangle$ (or mixed state $U\rho U^\dagger$). Some commonly used quantum gates are shown in Fig. 5. Then, a quantum circuit is a sequence of quantum gates: $(U_1, \ldots, U_d)$, where $d \geq 1$, and U_i $(1 \leq i \leq d)$ are quantum gates. We will use $U_{i:j}$ $(i \leq j)$ to denote the unitary matrix corresponding to the circuit slice from i-th gate to j-th gate; that is, $U_{i:j} = U_j \ldots U_{i+1}U_i$. In this work, we assume that the elementary gate set under consideration is composed of Clifford gates and Pauli rotation gates (i.e., $e^{i\theta P}$ where P is a Pauli operator and $\theta \in [-\pi, \pi]$, more

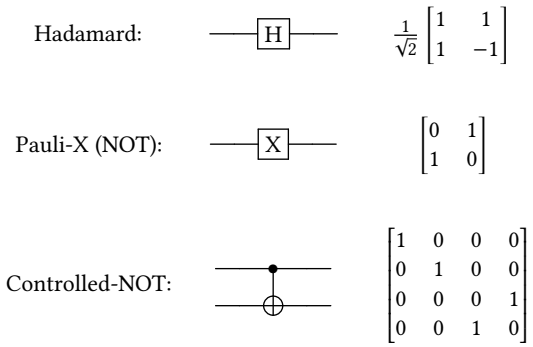


Fig. 5. Examples of single and multi-qubit quantum gates.

details can be found in Section 2.5). Detailed parameters of the benchmark circuits used in this paper are summarized in Table 1.

To extract classical information from a quantum state, we need to apply measurement on it. Mathematically, a quantum measurement can be described by a set of positive operators $M = \{M_i\}$, where $\sum_i M_i = I$. This set M is known as the Positive Operator-Valued Measure (POVM). When we perform a measurement M on a quantum state ρ, we observe an outcome i with probability $\mathrm{tr}(M_i\rho)$. Unlike the classical systems, the quantum states are generally disturbed after measurement, making it challenging to reuse the same quantum state for multiple measurements.

Example 1. *Consider the measurement on a qubit in the computational basis:* $\{M_0, M_1\}$*, where* $M_0 = |\omega\rangle\langle\omega|$*,* $M_1 = |\omega'\rangle\langle\omega'|$*, in which* $|\omega\rangle = \frac{1}{\sqrt{2}}e^{-i5\pi/16}|0\rangle + \frac{1}{\sqrt{2}}e^{i5\pi/16}|1\rangle$ *and* $|\omega'\rangle = \frac{1}{\sqrt{2}}e^{i3\pi/16}|0\rangle + \frac{1}{\sqrt{2}}e^{-i3\pi/16}|1\rangle$*. If we perform the measurement* $\{M_0, M_1\}$ *on a qubit in state* $|+\rangle\langle+|$*, where* $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$*, then we get outcome* 0 *with probability* $\mathrm{tr}(M_0|+\rangle\langle+|) = \cos^2(5\pi/16)$*, and get outcome* 1 *with probability* $\mathrm{tr}(M_1|+\rangle\langle+|) = \sin^2(5\pi/16)$.

2.2 Fault Models

Let $U_{\mathrm{CUT}} := (\tilde{U}_1, \ldots, \tilde{U}_d)$ be the circuit under test (CUT). Each gate $\tilde{U}_i$ is specified by its fault-free version U_i and has the potential to be faulty. We assume that its fault model U_i' is known in advance. Our goal is to determine whether the given CUT is fault-free or faulty. In this work, we adopt *the single-fault assumption* which is typically used in test generation and evaluation for both classical circuits [42] and quantum circuits [5, 34]: the cause of a circuit failure is attributed to only one faulty gate. This assumption is reasonable for logical (fault-tolerant) quantum circuits, since the effect of noise on single logical gate is fairly small such that the probability of single fault occurring dominates the probability of multiple faults occurring [19, 37].

Example 2. *Consider a simple example of the fault models. In the Quantum Fourier Transform (QFT) circuit (see Fig. 6a), the middle* Z*-axis-*$\pi/4$*-rotation gate has the potential to suffer from a missing gate fault. That is, the fault-free gate is* $U = R_Z(\frac{\pi}{4}) = \begin{bmatrix}e^{-i\pi/8} & 0\\ 0 & e^{i\pi/8}\end{bmatrix}$*, and its fault model is* $U' = I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$.

2.3 Discrimination of Quantum Gates

A basic step in quantum ATPG is to discriminate a gate U_i from its faulty version U_i'. This problem has been well studied by the quantum information community [3, 24, 25]. A general manner is by inputting a mixed state ρ to this unknown gate and then performing the measurement to obtain the result. Mathematically, the distinguishability between $U_i\rho U_i^\dagger, U_i'\rho U'^\dagger_i$, *i.e.*, the trace distance $tr(|U_i\rho U_i^\dagger - U_i'\rho U'^\dagger_i|)$ by the Holevo-Helstrom theorem [24, 25], must be maximized. Due to the convexity of trace distance, the optimal input can always be achieved by a pure state $\rho = |\psi\rangle\langle\psi|$. Thus the problem is equivalent to minimizing the inner product of $U_i|\psi\rangle$ and $U_i'|\psi\rangle$:

$$\min_{|\psi\rangle} \left|\langle\psi|U_i^\dagger U_i'|\psi\rangle\right|. \tag{2}$$

Let the spectral decomposition of $U_i^\dagger U_i'$ be $\sum_i \lambda_i|v_i\rangle\langle v_i|$, and $|\psi\rangle = \sum_i \alpha_i|v_i\rangle$. Then the optimization problem becomes:

$$\min_{|\psi\rangle} \left|\sum_i \lambda_i\langle\psi|v_i\rangle\langle v_i|\psi\rangle\right| = \min_{\sum|\alpha_i|^2=1} \left|\sum_i \lambda_i|\alpha_i|^2\right|, \tag{3}$$

which can be solved by quadratic programming.

After obtaining the optimal input state $|\psi\rangle$ from Equation (3), the optimal measurement for distinguishing between the pair $|\phi\rangle \coloneqq U_i|\psi\rangle,\ |\phi'\rangle \coloneqq U_i'|\psi\rangle$ of output states of the two gates can be achieved by their Helstrom measurement. More specifically, assume:

$$\langle\phi|\phi'\rangle = re^{i\theta}, \tag{4}$$

and define:

$$\begin{aligned} |\omega\rangle &= \frac{1}{\sqrt{2}}\left(\frac{|\phi\rangle + e^{-i\theta}|\phi'\rangle}{\||\phi\rangle + e^{-i\theta}|\phi'\rangle\|} + \frac{|\phi\rangle - e^{-i\theta}|\phi'\rangle}{\||\phi\rangle - e^{-i\theta}|\phi'\rangle\|}\right), \\ |\omega'\rangle &= \frac{1}{\sqrt{2}}\left(\frac{|\phi\rangle + e^{-i\theta}|\phi'\rangle}{\||\phi\rangle + e^{-i\theta}|\phi'\rangle\|} - \frac{|\phi\rangle - e^{-i\theta}|\phi'\rangle}{\||\phi\rangle - e^{-i\theta}|\phi'\rangle\|}\right). \end{aligned} \tag{5}$$

Then the Helstrom measurement is the POVM with measurement operators $|\omega\rangle\langle\omega|, |\omega'\rangle\langle\omega'|$, corresponding to the predictions "fault-free" and "faulty" respectively. The optimal success probability for distinguishing between faulty and fault-free gates in a single experiment is given by

$$\frac{1}{2} + \frac{1}{2}\sqrt{1-r^2} \in (\frac{1}{2}, 1], \tag{6}$$

where r is defined in Equation (4). Note that when the fault effect is small (e.g., subtle perturbations on the phase), r will be close to 1 so that the optimal strategy will not be much better than a random guess. In such cases, conducting a significantly larger number of repeated experiments becomes necessary to reliably detect the fault.

Example 3. *Consider the fault model given in Example 2. The optimal input state distinguishing between the faulty and fault-free gates, by solving optimization problem (3), is* $|\psi\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. *The outputs of fault-free and faulty gates on* $|\psi\rangle$ *are* $\frac{1}{\sqrt{2}}(e^{-i\pi/8}|0\rangle + e^{i\pi/8}|1\rangle)$ *and* $|+\rangle$, *respectively. Then, the Helstrom measurement, given by Equation (5), is exactly the measurement presented in Example 1, where* M_0, M_1 *correspond to "fault-free" and "faulty", respectively. Therefore, when the gate is faulty, our strategy will output "faulty" with probability* $\mathrm{tr}(M_1|+\rangle\langle+|) = \sin^2(5\pi/16)$ *in a single experiment, as shown in Example 1. In addition, if the gate is fault-free, simple calculation shows that our strategy will output "fault-free" also with probability* $\sin^2(5\pi/16)$. *Indeed,* $\sin^2(5\pi/16)$ *is the optimal probability of successfully distinguishing these gates in a single experiment.*

Example 4. *In Example 3, we have shown that the success probability for detecting the missing gate fault on $R_Z(\frac{\pi}{4})$ gate in a single experiment is $\sin^2(5\pi/16)$. To obtain an overall success probability ≥ 0.9, we need to repeat the experiment for 11 times (this can be seen by calculating the cumulative binomial probability). In fact, the test time complexity (number of repeated experiments) varies a lot on different gates, for example:*

- *Detecting the missing gate fault on H gate with success probability ≥ 0.9 requires only 1 experiment,*
- *Detecting the missing gate fault on $R_Z(\frac{\pi}{16})$ gate with success probability ≥ 0.9 requires 171 repeated experiments.*

In general, to amplify the success probability from $\frac{1}{2}+\frac{1}{2}\sqrt{1-r^2}$ (see Equation (6)) to 0.9, the test time complexity (number of repeated experiments) is $\Theta(\frac{1}{1-r^2})$.

2.4 Test Patterns for Single-Fault Quantum Circuits

The above discussion considers gate-level discrimination. Now we move on to consider the circuit-level discrimination between the faulty and fault-free circuits. It is impractical to solve the optimization problem (3) at the full-circuit level due to the high computational complexity. Specifically, it requires $O(d \cdot N^3)$ arithmetic operations to first compute the matrix representing the whole circuit, and then $O(N^3)$ arithmetic operations to solve the optimization problem (3), where d is the number of gates and $N = 2^n$ is the dimension of the whole quantum system. Nevertheless, since there is at most one faulty gate in the circuit (single-fault assumption, see Section 2.2), we can locally solve the discrimination problem for the faulty and fault-free gates at the fault site, which generally act on a small sub-system (say, a single qubit or two qubits), obtaining the results $|\psi\rangle, |\omega\rangle, |\omega'\rangle$ as in Equations (3) and (5). Then we propagate the fault excitation signal $|\psi\rangle$ and fault effect $|\omega\rangle/|\omega'\rangle$ to the beginning and end of the circuit respectively. Formally, we have:

Proposition 1. *If quantum state ρ and measurement $\{M, I-M\}$ satisfy the following conditions:*

$$\begin{aligned} \rho &\sqsubseteq U_{1:i-1}^{\dagger}\left(Q \otimes |\psi\rangle\langle\psi|\right) U_{1:i-1}, \\ U_{i+1:d}\left(Q \otimes |\omega\rangle\langle\omega|\right) U_{i+1:d}^{\dagger} &\sqsubseteq M, \\ U_{i+1:d}\left(Q \otimes |\omega'\rangle\langle\omega'|\right) U_{i+1:d}^{\dagger} &\sqsubseteq I-M, \end{aligned} \tag{7}$$

where $\sqsubseteq$ is the Loewner order and Q is some projection on the state space of the qubits that are not in gates U_i and U_i', then ρ and $\{M, I-M\}$ are optimal state and measurement at the primary input and output for distinguishing between faulty and fault-free circuits, i.e., optimal in test time complexity.

We call the pair (ρ, M) satisfying constraints (7) a test pattern of CUT C with respect to its faulty version C_i. Note that unlike the optimization problem in previous section, there is flexibility in defining the optimal test pattern. This is because the parts of system other than the fault site can be arbitrary without affecting optimality. However, for simplicity, we can choose the test pattern (ρ, M) to be

$$\begin{aligned} \rho &= U_{1:i-1}^{\dagger}\Big(\frac{Q}{\mathrm{tr}(Q)} \otimes |\psi\rangle\langle\psi|\Big) U_{1:i-1} \\ M &= U_{i+1:d}\big(Q \otimes |\omega\rangle\langle\omega|\big) U_{i+1:d}^{\dagger}. \end{aligned} \tag{8}$$

Note that Equation (8) also involves matrices of exponential sizes, which is hard to directly evaluate. More specifically, if we choose $Q = |\phi\rangle\langle\phi|$, the computational complexity is $O(d \cdot N^2)$ (since it only requires matrix-vector multiplications), which is better than the full-circuit level optimization but is still inefficient. This issue is further alleviated in Sections 4.1 and 4.2, where several acceleration techniques are proposed. There, we use stabilizer projector decomposition (SPD)

to represent the the test pattern ($|\psi\rangle\langle\psi|, |\omega\rangle\langle\omega|$), and sequentially propagate the SPD to the primary input and output of the circuit.

Example 5. *Consider the fault model given in Example 2. The optimal state $|\psi\rangle$ and measurement $\{|\omega\rangle\langle\omega|, |\omega'\rangle\langle\omega'|\}$ for distinguishing the fault-free and faulty gates are given in Example 3 and Example 1. Then, after applying Equation (8), we obtain a test pattern (ρ, M), where $\rho = \frac{1}{4}\cdot I\otimes|0\rangle\langle 0|\otimes I$ and $M = I\otimes\begin{bmatrix} 0.5 & 0 & a & b \\ 0 & 0.5 & b & a \\ a^* & b^* & 0.5 & 0 \\ b^* & a^* & 0 & 0.5 \end{bmatrix}$, in which $a \approx -0.135-0.326i, b \approx 0.326 - 0.135i$. Note that M is a intricate measurement, which lacks efficient and robust implementations. Our proposed method can decompose this challenging test pattern into several simpler test patterns, which can be efficiently implemented using the robust Clifford-only circuits. The produced results are shown in Fig. 6.*

2.5 Stabilizer Formalism and Clifford Circuits

As discussed in Section 1, one major challenge lies in the test equipment complexity and the robustness of test application. More specifically, the difficulty arises from the hardness of implementing the state ρ and measurement M in Equation (8). Our solution to this problem leverages the stabilizer formalism and the associated properties of Clifford circuits, which are commonly employed in fault-tolerant quantum computing. For the reader's convenience, we provide a brief review of these concepts in the following subsection.

2.5.1 Pauli Operators. Pauli operators are widely used in quantum computation and quantum information [33]. Single-qubit Pauli operators consist of I, X, Y, Z where

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{9}$$

Any n-qubit Pauli operator P is of the form: $P_1 \otimes P_2 \otimes \cdots \otimes P_n$ where each P_i is chosen from I, X, Y, Z. We will use X_i (or Y_i, Z_i) to denote the Pauli operator $P_1 \otimes \cdots \otimes P_n$ with $P_i = X$ (or Y, Z) and $P_j = I$ for all $j \neq i$. Suppose P is a Pauli operator, we will use $\underline{e^{i\theta}P}$ to denote the Pauli operator by simply discarding the phase $e^{i\theta}$, (*e.g.* $\underline{ZX} = \underline{iY} = Y$). We call $e^{i\theta}P$ a *signed* Pauli operator, if $\theta \in \{0, \pi\}$ and P is a Pauli operator. Without causing confusion, the same letters (*e.g.* P, Q) may be used to denote either Pauli or signed Pauli operators. Note that any two (signed) Pauli operators P, Q either commute (*i.e.*, $PQ = QP$) or anti-commute (*i.e.*, $PQ = -QP$).

There is a very useful representation for the Pauli operators, using the vectors over GF(2). In this paper, we use $00, 01, 10, 11$ to represent I, Z, X, Y respectively. The Pauli operator $P = P_1 \otimes \ldots \otimes P_n$ is then represented by the vector $v(P) = (v_1, \ldots, v_{2n})$, where v_{2i-1}, v_{2i} corresponds to the representation of P_i. Note that the vector addition coincides with the Pauli operator multiplication up to a global phase. That is, if $\underline{PQ} = R$ where P, Q, R are Pauli operators, then $v(P) + v(Q) = v(R)$. For a signed Pauli operator P, we simply define its vector representation as $v(\underline{P})$. We say that a set of (signed) Pauli operators are *independent* if their vector representations are independent.

2.5.2 Stabilizer Formalism. We say that a state $|\psi\rangle$ is *stabilized* by a signed Pauli operator P if $P|\psi\rangle = |\psi\rangle$. In this case, P is called the *stabilizer* of $|\psi\rangle$. The basic idea of the stabilizer formalism is that many quantum states can be more easily described by their stabilizers [33]. In general, suppose G is a group of n-qubit signed Pauli operators and $-I \notin G$. We call G the *stabilizer group*. Let V be the set of all n-qubit states which are stabilized by every element of G. Then V forms a vector space. The projector S onto V is called the *stabilizer projector* corresponding to group G. When V is

one-dimensional, its projector S is of the form $S = |\phi\rangle\langle\phi|$ and $|\phi\rangle$ is called the *stabilizer state* corresponding to G. A set of elements $P_1, \ldots, P_l$ is said to be the *generators* of G if every element of G can be written as a product of elements from the list $P_1, \ldots, P_l$, and we write $G = \langle P_1, \ldots, P_l\rangle$. There are some useful properties of the stabilizer formalism. First, the elements of G commute, *i.e.*, for any $P, Q \in G$, $PQ = QP$. Second, the stabilizer projector S can be represented by

$$S = \prod_{i=1}^{l} \frac{I + P_i}{2} = \frac{1}{|G|} \sum_{P_i \in G} P_i \tag{10}$$

where $P_1, \ldots, P_l$ are the generators of G.

2.5.3 Clifford Operation. Suppose we apply a unitary U to the stabilizer space V. Let $|\phi\rangle \in V$. Then for any $g \in G$: $U|\phi\rangle = Ug|\phi\rangle = UgU^\dagger U|\phi\rangle$ which means that $U|\phi\rangle$ is stabilized by $UgU^\dagger$. Thus subspace UV is stabilized by the group $UGU^\dagger \equiv \{UgU^\dagger | g \in G\}$ and the corresponding stabilizer projector is $USU^\dagger$. For a unitary U, if it takes signed Pauli operators to signed Pauli operators under conjugation, *i.e.*, $UPU^\dagger = P'$, then we call it a *Clifford operation*. This transformation takes on a particularly appealing form. That is, for any stabilizer projector S, $USU^\dagger = S'$ where S' is also a stabilizer projector. It was proved that any n-qubit Clifford operation can be implemented using Hadamard, Phase and CNOT gates, with the circuit size no more than $O(n^2/\log n)$ [2]. Clifford circuits can be efficiently simulated on a classical computer according to the Gottesman-Knill theorem [16]. Furthermore, using fault-tolerant techniques based on CSS codes, including the popular Steane code and surface codes, they can be efficiently implemented with little error rate [14, 44]. This contrasts with the non-Clifford gates (*e.g.* T gate), which incur more than a hundred times cost for fault-tolerant implementation. The following facts about Clifford operations will be needed in this paper:

Lemma 1. *Given two lists of signed Pauli operators $(P_1, \ldots, P_n)$ and $(Q_1, \ldots, Q_n)$. If*

(1) $\forall i, j \leq n, P_i, P_j$ *commute* $\longleftrightarrow Q_i, Q_j$ *commute,*
(2) $(P_1, \ldots, P_n)$ *and* $(Q_1, \ldots, Q_n)$ *are both independent*

then there is a Clifford unitary U with $UP_iU^\dagger = Q_i$ for all i.

Corollary 1. *For any stabilizer projector A of rank 2^m in an n-qubit system, there is a Clifford circuit U_A such that $U_A(|0\rangle\langle 0|^{\otimes n-m} \otimes I)U_A^\dagger = A$.*

3 SPD-BASED SAMPLING ALGORITHM FOR TEST APPLICATION

Test application is the process of applying a test pattern obtained from test generation to the CUT and analyzing the output responses [42]. More specifically, given a quantum test pattern (ρ, M), the test application includes preparing the input quantum state ρ, and performing measurement $\{M, I - M\}$ on the output of CUT to obtain measurement results. By collecting measurement results from repeated experiments, we can estimate the value of $\mathrm{tr}(MU_{\mathrm{CUT}}\rho U_{\mathrm{CUT}}^\dagger)$, which represents the probability of the CUT passing the test. Consequently, if $\mathrm{tr}(MU_{\mathrm{CUT}}\rho U_{\mathrm{CUT}}^\dagger) > 0.5$, we claim that the CUT is fault-free, otherwise faulty.

In this section, we present the stabilizer projector decomposition (SPD) and an SPD-based sampling algorithm for test application. Our algorithm adopts Clifford circuits as test equipment, which is robust against SPAM error and has better test equipment complexity.

3.1 Stabilizer Projector Decomposition

As mentioned in Subsection 2.5, Clifford operations have reliable and efficient fault-tolerant implementation. We choose to use Clifford-only circuit to realize our test application. However, a test pattern (ρ, M) as a solution of Equation (7) is generally non-Clifford. To handle this problem, we introduce the stabilizer projector decomposition (SPD), which generalizes the notion of pseudo-mixtures for stabilizer states [27].

Definition 1 (Stabilizer Projector Decomposition, SPD). *Let $0 \sqsubseteq A \sqsubseteq I$. Then a stabilizer projector decomposition (*SPD *for short) of A is a finite set $\mathcal{A} = \{(a_i, A_i)\}_i$ such that*

$$A = \sum_i a_i A_i,$$

where A_i are stabilizer projectors and a_i are real numbers.

We will use the calligraphic letters (*e.g.* $\mathcal{A}, \mathcal{B}$) to denote the SPDs and use $\mathrm{SPD}(A)$ to denote the set of all possible SPDs of operator A. For convenience, given an SPD $\mathcal{A} = \{(a_i, A_i)\}$, we use $Q \otimes \mathcal{A}$ to denote the SPD $\{(a_i, Q \otimes A_i)\}$, where Q is a stabilizer projector. Similarly, we write $U\mathcal{A}U^\dagger$ for the SPD $\{(a_i, UA_iU^\dagger)\}$ and $c\mathcal{A}$ for the SPD $\{(c \cdot a_i, A_i)\}$ where U is a Clifford unitary, and c is a real number. We also define:

$$\mathrm{tr}(\mathcal{A}) = \mathrm{tr}(\sum_i a_i A_i).$$

In our SPD-based sampling algorithm, the following norms are needed.

Definition 2 (Norms of SPD). *The 1-norm ν and weighted 1-norm ν^* of SPD $\mathcal{A} = \{(a_i, A_i)\}$ are defined by:*

$$\begin{aligned} \nu(\mathcal{A}) &= \sum_i |a_i| \\ \nu^*(\mathcal{A}) &= \sum_i |a_i| \mathrm{tr}A_i \end{aligned} \tag{11}$$

3.2 SPD-based Sampling Algorithm

Now we turn to the main problem in this section: estimate the expectation $\mathrm{tr}(MU_{\mathrm{CUT}}\rho U_{\mathrm{CUT}}^\dagger)$ for a given test pattern (ρ, M). Here we assume that the SPDs $\mathcal{A}, \mathcal{B}$ for ρ, M are provided, respectively:

$$\begin{aligned} \rho &= \sum_i a_i A_i, \\ M &= \sum_i b_i B_i \end{aligned} \tag{12}$$

but leave the problem of generating them to the next section.

First, we can rewrite the expectation under estimation in terms of the given SPDs:

$$\begin{aligned} \mathrm{tr}(MU_{\mathrm{CUT}}\rho U_{\mathrm{CUT}}^\dagger) &= \sum_{ij} a_i b_j \mathrm{tr}(B_j U_{\mathrm{CUT}} A_i U_{\mathrm{CUT}}^\dagger) \\ &= \sum_{ij} a_i b_j \mathrm{tr}A_i \mathrm{tr}(B_j U_{\mathrm{CUT}} \frac{A_i}{\mathrm{tr}A_i} U_{\mathrm{CUT}}^\dagger) \end{aligned} \tag{13}$$

For simplicity, we put:

$$\mathrm{tr}_{ij} = \mathrm{tr}(B_j U_{\mathrm{CUT}} A_i / \mathrm{tr}A_i U_{\mathrm{CUT}}^\dagger).$$

Since A_i, B_j are stabilizer projectors, by Corollary 1, we can implement the state preparation of $A_i/\mathrm{tr}A_i$ and projective measurement of B_j using Clifford-only circuits. Then, we can sample from the Bernoulli distribution of parameter tr_{ij}, by applying the CUT on input state $A_i/\mathrm{tr}A_i$ and sampling the result according to measurement B_j.

To further estimate $\sum_{ij} a_i b_j \mathrm{tr}A_i \mathrm{tr}_{ij}$, we use the quasi-probability distribution [27, 35] $\{a_i b_j \mathrm{tr}A_i\}_{ij}$ to form a true probability distribution $\{p_{ij}\}_{ij}$, where

$$p_{ij} = |a_i b_j \mathrm{tr}A_i| / \sum_k |a_k \mathrm{tr}A_k| \sum_k |b_k|.$$

In each single trial, we sample i, j from this probability distribution p_{ij} and then sample a result $m \in \{0, 1\}$ from the Bernoulli distribution of parameter tr_{ij}. We do not record m but its corrected version:

$$\begin{aligned} \hat{m} &= m \cdot \mathrm{sign}(a_i b_j)(\sum_k |a_k| \mathrm{tr}A_k)(\sum_k |b_k|) \\ &= m \cdot \mathrm{sign}(a_i b_j) \nu^*(\mathcal{A}) \nu(\mathcal{B}) \end{aligned} \tag{14}$$

where $\mathrm{sign}(x) = 1$ if $x \geq 0$ and $\mathrm{sign}(x) = -1$ otherwise. Note that $\hat{m}$ is essentially sign-corrected by $\mathrm{sign}(a_i b_j)$ and modulus-corrected by $\nu^*(\mathcal{A})\nu(\mathcal{B})$. The expected value of random variable $\hat{m}$ is

$$\begin{aligned} E(\hat{m}) &= \sum_{ij} p_{ij} \mathrm{tr}_{ij} \mathrm{sign}(a_i b_j)(\sum_k |a_k| \mathrm{tr}A_k)(\sum_k |b_k|) \\ &= \sum_{ij} a_i b_j \mathrm{tr}A_i \mathrm{tr}_{ij} = \mathrm{tr}(M U_{\mathrm{CUT}} \rho U_{\mathrm{CUT}}^\dagger) \end{aligned} \tag{15}$$

This means $\hat{m}$ can serve as an unbiased estimator of the target value $\mathrm{tr}(M U_{\mathrm{CUT}} \rho U_{\mathrm{CUT}}^\dagger)$. By repeating this sampling process, we can estimate the result to any desired accuracy, where the number of repetition depends on the variance of $\hat{m}$. The details of this SPD-based sampling algorithm are summarized in Algorithm 1.

Then, we analyze the performance of Algorithm 1. Note that $\hat{m} \in [-\nu^*(\mathcal{A})\nu(\mathcal{B}), \nu^*(\mathcal{A})\nu(\mathcal{B})]$. By the Hoeffding's inequality, we know that

$$\frac{2}{\delta^2} \ln(\frac{2}{\epsilon}) \left[\nu^*(\mathcal{A})\nu(\mathcal{B})\right]^2 \tag{16}$$

samples are needed to estimate the result with error less than δ and success probability exceeding $1 - \epsilon$. The above result is summarized in Theorem 1.

Theorem 1. *Algorithm 1 estimates* $\mathrm{tr}(M U_{CUT} \rho U_{CUT}^\dagger)$ *to additive error* δ *with probability larger than* $1 - \epsilon$*, and has test time complexity (i.e., the number of experiments on the CUT):*

$$O\left(\frac{\ln(1/\epsilon)}{\delta^2} \left[\nu^*(\mathcal{A})\nu(\mathcal{B})\right]^2\right). \tag{17}$$

Note that the term $\left[\nu^*(\mathcal{A})\nu(\mathcal{B})\right]^2$ corresponds to the additional overhead in test time complexity, introduced by our SPD-based sampling algorithm. As mentioned in Section 1.3, this can be viewed as a trade off between quantum resources (i.e., test equipment complexity) and classical resources (i.e., test time complexity). Specifically, our SPD-based sampling algorithm provides efficiency in test equipment complexity (to be justified in Table 3) and robustness in test application (due to the robustness of Clifford circuits), while it also requires more repeated experiment to obtain a valid test result (due to the uncertainty induced by the classical randomness). In the next section, we will focus on reducing the test time complexity, as well as the computational complexity of the test generation process.

Algorithm 1 SPD-based sampling algorithm

Input: SPD for ρ, M (see Equation (12)), Quantum CUT U_{CUT}, required precision δ and success probability $1-\epsilon$
Output: An estimation m of $\text{tr}(MU_{\text{CUT}}\rho U_{\text{CUT}}^\dagger)$

Initialize the estimation result $m \leftarrow 0$.
Set the iteration number $T \leftarrow \frac{2}{\delta^2}\ln(\frac{2}{\epsilon})[\,\nu^*(\mathcal{A})\nu(\mathcal{B})\,]^2$
for $t = 1, 2, \ldots, T$ **do**
  Sample values i and j from distribution $|a_i|\text{tr}A_i/\nu^*(\mathcal{A})$ and $|b_j|/\nu(\mathcal{B})$ respectively.
  Uniformly sample value $l \in \{0, \ldots, \text{tr}A_i - 1\}$
  Initialize the first $n - \log \text{tr}A_i$ qubits to $|0\rangle$, and last $\log \text{tr}A_i$ qubits to $|l\rangle$.
  Apply the Clifford circuit U_{A_i}. ▷ Corollary 1
  Apply the CUT U_{CUT}.
  Apply the Clifford circuit $U_{B_j}^\dagger$. ▷ Corollary 1
  Measure the first $n - \log \text{tr}B_j$ quibts on computational basis.
  if The measurement result is $|0\rangle$ **then**
    $m \leftarrow m + \text{sign}(a_i b_j)\nu^*(\mathcal{A})\nu(\mathcal{B})$
  end if
end for
$m \leftarrow m/T$
return m

4 SPD GENERATION

In the previous section, an algorithm for robust test application was presented under the assumption that the SPDs of a test pattern (ρ, M) are given. This section is devoted to solving the remaining problem, namely the SPD generation. Our aim is to optimize the test time complexity induced by the generated SPDs, while also reduce the computational complexity of the generation process.

As shown in Theorem 1, we want to generate the SPDs that have as low ν and ν^* values as possible. In this sense, given a positive operator A, one can find the ν-optimal SPD by solving this convex optimization problem:

$$\begin{aligned} \min_{x} \quad & \|x\|_1 \\ \text{subject to} \quad & \mathcal{S}x = b \end{aligned} \tag{18}$$

where $\mathcal{S}_{ij} = \text{tr}(P_i S_j)$, $b_i = \text{tr}(P_i A)$, P_i is the i-th Pauli operator and S_j is the j-th stabilizer projector. An optimal solution x of problem (18) indicates a ν-optimal SPD $\{(x_i, S_i)\}$ for A. For computational efficiency, we can discard those (x_i, S_i) with $x_i = 0$. Similarly, we can obtain the ν^*-optimal SPD by solving this convex optimization problem:

$$\begin{aligned} \min_{x} \quad & \|w \circ x\|_1 \\ \text{subject to} \quad & \mathcal{S}x = b \end{aligned} \tag{19}$$

where $w_i = \text{tr}(S_i)$ and $\circ$ denotes the element-wise product.

Now, our goal is to solve the SPD-optimization problems (19) and (18) for the test pattern ρ and M obtained in (8), respectively. As noted in Section 2.4, ρ and M can be obtained more efficiently through a fault propagation strategy. We adopt a similar idea here, embedding the SPD-optimization in the fault propagation process. That is, at the very beginning, we solve the SPD-optimization problems locally for the initial test pattern at the fault site, and use the resulting SPDs to represent the initial test pattern. Then, we forward (or backward) propagate the fault effect (or excitation signal) represented by the SPD to the primary output (or primary input) of the circuit, as with the D-algorithm. More specifically, the propagation process is as follows:

Algorithm 2 SPD generation

Input: Description of the fault-free circuit $(U_1, \ldots, U_d)$, potential fault position i and corresponding fault model U_i'.
Output: SPD for ρ and M, where ρ and M is the optimal input and measurement that can detect the fault.

Let $(|\psi\rangle, |\omega\rangle)$ be the test pattern for distinguishing U_i and U_i'. ▷ see Section 2.3
Let $\mathcal{A}$ be the ν^*-optimal SPD for $|\psi\rangle\langle\psi|$ ▷ Problem (19)
Let $\mathcal{B}$ be the ν-optimal SPD for $|\omega\rangle\langle\omega|$ ▷ Problem (18)
$\mathcal{A} \leftarrow I \otimes \mathcal{A}$, $\mathcal{B} \leftarrow I \otimes \mathcal{B}$ ▷ Padding (choosing $Q = I$ in Equation (8))
for $j = i - 1$ down to 1 **do** ▷ Backward propagation
    if U_j is Clifford **then**
        $\mathcal{A} \leftarrow U_j^\dagger \mathcal{A} U_j$
    else
        Let U_C = LOC_EXPL$(U_j^\dagger, \mathcal{A})$ ▷ Algorithm 3
        $\mathcal{A} \leftarrow$ the optimal SPD computed through diagram (20)
    end if
end for
$\mathcal{A} \leftarrow \mathcal{A}/\mathrm{tr}(\mathcal{A})$
for $j = i + 1$ to d **do** ▷ Forward propagation
    if U_j is Clifford **then**
        $\mathcal{B} \leftarrow U_j \mathcal{B} U_j^\dagger$
    else
        Let U_C = LOC_EXPL$(U_j, \mathcal{B})$ ▷ Algorithm 3
        $\mathcal{B} \leftarrow$ the optimal SPD computed through diagram (20)
    end if
end for
return $\mathcal{A}$, $\mathcal{B}$

- Suppose at the current propagation step we have at hand a positive operator A representing the current outpu pattern M (or input pattern ρ), and also an SPD $\mathcal{A}$ for A. Suppose the next gate encountered is U.
- We re-solve the optimization problem (19) (or problem (18)) to find a new SPD for $UAU^\dagger$ (or $U^\dagger AU$).

However, it is practically intractable to directly deal with these problems since the number of stabilizer projectors grows exponentially [17]. To alleviate this issue, several acceleration techniques are employed in each propagation step, such as the locality exploiting and sparsity exploiting.

4.1 Locality Exploiting

In this section, we introduce the locality exploiting technique in the process of SPD propagation. The idea is, **1)** using a Clifford unitary U_C to map the current SPD to a local form, **2)** solving problem (18) or (19) in the "non-trivial" subsystem, and **3)** mapping back. The optimality is preserved thanks to the nice properties of ν (and also ν^* analogously), *i.e.*, the Clifford invariance (C-Inv) and tensor invariance (T-Inv):

(1) **C-Inv**: For any Clifford unitary U_C, if an SPD $\mathcal{A}$ is ν-optimal (ν^*-optimal), then $U_C \mathcal{A} U_C^\dagger$ is also ν-optimal (ν^*-optimal).
(2) **T-Inv**: For any stabilizer projector S, if an SPD $\mathcal{A}$ is ν-optimal (ν^*-optimal), then $S \otimes \mathcal{A}$ is also ν-optimal (ν^*-optimal).

Specifically, suppose we need to find the v-optimal (or v^*-optimal) SPD for $UAU^\dagger$, and A is represented by an SPD $\mathcal{A}$, then our strategy follows this diagram:

$$\begin{array}{ccc} (U,\mathcal{A}) & \xrightarrow{\text{find optimal SPD}} & U_C^\dagger(U_\perp A_\perp U_\perp^\dagger \otimes \mathcal{A}'')U_C \\ \downarrow U_C & & \uparrow \text{C-Inv} \\ (U_\perp \otimes U', A_\perp \otimes \mathcal{A}') & & U_\perp A_\perp U_\perp^\dagger \otimes \mathcal{A}'' \\ \downarrow \text{localize} & & \uparrow \text{T-Inv} \\ (U',\mathcal{A}') & \xrightarrow{\text{solve problem (18) or (19) for } U'A'U'^\dagger} & \mathcal{A}'' \end{array} \tag{20}$$

where U_C is a Clifford unitary such that $U \xrightarrow{U_C} U_\perp \otimes U'$ and $\mathcal{A} \xrightarrow{U_C} A_\perp \otimes \mathcal{A}'$ by conjugation, where $U_\perp$, $A_\perp$ are Clifford unitary and stabilizer projector, which are the "trivial" parts in our computation.The above procedures are summarized in Algorithm 2.

The remaining problem is how to find a good U_C that satisfies diagram (20). We propose an algorithm (shown in Algorithm 3) for finding such U_C that reveals as much locality as possible. Similar to [38], we assume that the elementary gate set under consideration is the set of Clifford + Pauli rotations (i.e., $e^{i\theta P}$ where P is a Pauli operator and $\theta \in [-\pi, \pi]$). Note that this form covers some important and practical universal quantum gate sets, such as Clifford + T gate set, or more generally the Clifford + Z rotation gate set.

We provide here the main intuition behind Algorithm 3 (for simplicity, here we only consider $\mathcal{A}$, without U), and the detailed analyses are presented in the Appendix. First note that any stabilizer projector is equivalent to $|0\rangle\langle 0| \otimes I$ under Clifford conjugation. Thus, to extract a stabilizer projector $A_\perp$ from the SPD $\mathcal{A}$, Algorithm 3 works roughly as follows,

(1) extract a rank-1 projector $|0\rangle\langle 0|$ (steps 3-8), by computing the intersection of all components in $\mathcal{A}$,
(2) extract a full rank projector I (steps 9-20), by computing the complement of the union of all components in $\mathcal{A}$,
(3) combine previous results (step 21).

Finally, the algorithm can exploit a smaller "non-trivial" subsystem of size $(s+t)/2$ (see Algorithm 3 for details). Its performance is theoretically guaranteed by the following theorem (the proof is deferred to the Appendix):

Theorem 2. *Algorithm 3 finds a Clifford unitary U_C that satisfies diagram (20). Moreover, if either of the following two conditions is satisfied,*

(1) *$\mathcal{A}$ is v-optimal and has the smallest v^* among all other v-optimal* SPDs *for A*
(2) *$\mathcal{A}$ is v^*-optimal and has the smallest v among all other v^*-optimal* SPDs *for A*

then U_C is optimal (i.e., reveals the minimum "non-trivial" subsystem) among all Clifford unitaries that satisfy diagram (20).

Theorem 2 states that our locality exploiting algorithm can always leverage some locality (i.e., convert the problem into a subsystem), and under certain mild conditions, it can maximize locality exploitation (i.e., convert the problem into a subsystem with the smallest possible size). Both conditions in Theorem 2 can be achieved by a strategy named multi-objective optimization, more details are shown in section 4.2.1. However, it should be noted that, in practice, the system size might be too large to solve the multi-objective optimization problem, so that the conditions in Theorem 2 may not hold at all times. Even worse, there might be no locality to exploit in some cases. To alleviate this issue, we also propose other acceleration techniques such as sparsity exploiting (see Section 4.2.2) and Clifford channel decomposition (see Section 4.2.3).

Algorithm 3 Locality exploiting, LOC_EXPL$(U, \mathcal{A})$

Input: Non-Clifford U, SPD $\mathcal{A} = \{(a_i, A_i)\}_{i=1}^m$.
Output: Clifford U_C that satisfies diagram (20).

Suppose $U = e^{i\theta P_U}$, P_U is a Pauli operator and $|\theta| \leq \pi$
Let G_i be the group associated to stabilizer projector A_i
Let $\hat{P}_1, \ldots, \hat{P}_l$ be the independent generators of $G_1 \cap \ldots \cap G_m$
if P_U anti-commutes with some $\hat{P}_i$ **then**
    Let P_U only anti-commutes with $\hat{P}_l$ (exchange the subscripts of $\hat{P}_i$ or set $\hat{P}_i \leftarrow \hat{P}_i\hat{P}_j$ if necessary)
    $l \leftarrow l - 1$
end if
Generate the Clifford unitary V_1 that maps $\hat{P}_1, \ldots, \hat{P}_l, P_U$ to $Z_1, \ldots, Z_l, Z_{l+1}$ ▷ Lemma 1
For each $V_1 G_i V_1^\dagger$, find a set of independent generators of the form $Z_1, \ldots, Z_l, I_l \otimes P_{i,1}, \ldots, I_l \otimes P_{i,l_i}$
Find a maximal independent set $Q_1, \ldots, Q_s$ of $Z_1, P_{1,1}, \ldots, P_{1,l_1}, \ldots, P_{m,1}, \ldots, P_{m,l_m}$
for $i \leq s, i < j \leq s$ **do**
    if Q_i, Q_j anti-commute **then**
        for $k \leq s, k \neq i, j$ **do**
            If Q_k, Q_i anti-commute, then $Q_k \leftarrow \underline{Q_k Q_j}$
            If Q_k, Q_j anti-commute, then $Q_k \leftarrow \underline{Q_k Q_i}$
        end for
    end if
end for
Exchange the subscripts of $Q_1, \ldots, Q_s$ so that $Q_1, \ldots, Q_t$ commute with all $Q_i, i \leq s$ and Q_{t+2i-1} only anti-commutes with Q_{t+2i} for $i = 1, \ldots, (s-t)/2$
Generate a Clifford unitary V_2 that maps $Q_1, \ldots, Q_t$ to $Z_1, \ldots, Z_t$ and maps $Q_{t+1}, \ldots, Q_s$ to $Z_{t+1}, X_{t+1}, \ldots, Z_{(t+s)/2}, X_{(t+s)/2}$ ▷ Lemma 1
$V \leftarrow V_2 V_1$
$U_C := V$
return U_C

4.2 Implementation Details

The algorithm presented above still faces some practical challenges in its implementation. In this subsection, we propose several important techniques to meet these challenges and thus improve the feasibility of the algorithm.

4.2.1 Multi-Objective Optimization. To meet the requirements of Theorem 2, we need to obtain an SPD that is primarily ν-optimal and secondarily ν^*-optimal in the forward propagation (or primarily ν^*-optimal and secondarily ν-optimal in the backward propagation). Consider the case of forward propagation. Let $\overline{\theta}$ be the optimum value of the optimization problem (18) and define the secondary optimization problem:

$$\begin{aligned} \min_x \quad & \|\omega \circ x\|_1 \\ \text{subject to} \quad & \mathcal{S}x = b, \ \|x\|_1 \leq \overline{\theta} \end{aligned} \tag{21}$$

Then, the solution of Problem (21) is primarily ν-optimal and secondarily ν^*-optimal. In practice, we use the perturbation method [31] to compute the solution at once. Consider the following problem:

$$\begin{aligned} \min_x \quad & \|x\|_1 + \epsilon\|\omega \circ x\|_1 \\ \text{subject to} \quad & \mathcal{S}x = b \end{aligned} \tag{22}$$

It can be shown that, for sufficiently small ϵ, the optimal solution x of Problem (22) is also the optimal solution of Problem (21). For the case of backward propagation, we have a similar conclusion.

4.2.2 Sparsity Exploiting. Given an SPD $\mathcal{A} = \{(a_i, A_i)\}_{i=1}^{k}$, we say that $\mathcal{A}$ is k-sparse as it has k non-zero terms in the decomposition. Since $\mathcal{A}$ is represented by k pairs of real number and stabilizer projector, any non-trivial operation on $\mathcal{A}$ has time complexity at least $\Omega(k)$. Besides, in the worst case, k may grow exponentially with the number of non-Clifford gates applied on this SPD. To accelerate our algorithm, we exploit the sparsity of $\mathcal{A}$ using the iterative log heuristic method [29]. This method finds a sparse solution in the feasible set by finding the local optimal point of the logarithmic function $\sum_i \log(\delta + |x_i|)$. As this objective function is not convex, a heuristic method is applied where we first replace the objective by its first-order approximation, then solve and re-iterate.

For efficiency, we only use the stabilizer projectors that already exist in $\mathcal{A}$ as the basis, instead of using all stabilizer projectors. We also require the solution is primarily ν-optimal and secondarily ν^*-optimal (in the forward propagation for example). Combining the perturbation method [31] mentioned earlier, we obtain the following optimization problem:

$$\begin{aligned} \min_{x} \quad & \|x\|_1 + \epsilon\|\omega \circ x\|_1 + \epsilon^2\|L \circ x\|_1 \\ \text{subject to} \quad & \mathcal{S}x = b \end{aligned} \tag{23}$$

where $\mathcal{S}_{ij} = \mathrm{tr}(P_i A_j)$, P_i is the i-th Pauli operator, A_j is the j-th stabilizer projector in SPD $\mathcal{A}$, $b_i = \mathrm{tr}(P_i A)$ and the weight vector L is re-adjusted in each iteration based on the rule:

$$L_i = 1/(\gamma + |x_i|) \tag{24}$$

where γ is a small threshold value and the iteration number is fixed to 5 for simplicity. Note that in problem (23), the primary, secondary and tertiary objectives are ν, ν^* and sparsity respectively and this technique also applies to the backward propagation similarly. We use this technique to make the SPD sparse when a non-Clifford gate is applied in the forward or backward propagation.

4.2.3 Clifford Channel Decomposition. If the system size after exploiting locality and sparsity is still large, we can use the Clifford channel decomposition [4] to directly compute the sub-optimal SPD. For example, we use $\boldsymbol{Z}(\theta)$ (bold Z) to denote the quantum channel corresponding to the coherent rotation $\rho \mapsto Z(\theta)\rho Z(\theta)^\dagger$, where $Z(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$. Then $\boldsymbol{Z}(\theta)$ can be decomposed as

$$\boldsymbol{Z}(\theta) = \frac{1 + \cos\theta - \sin\theta}{2}\boldsymbol{I} + \frac{1 - \cos\theta - \sin\theta}{2}\boldsymbol{Z} + \sin\theta\boldsymbol{S} \tag{25}$$

where I denotes the identity channel, $\boldsymbol{Z}$ and $\boldsymbol{S}$ denote the channel $\rho \mapsto Z\rho Z^\dagger$ and channel $\rho \mapsto S\rho S^\dagger$, respectively. The numerical results demonstrate that this decomposition is nearly optimal for $0 \le \theta \le \pi$, *i.e.*, it has the nearly smallest 1-norm of coefficients among all Clifford channel decomposition (however, this does NOT imply the near-optimality of the resulted SPD). This decomposition can also be generalized to any Pauli rotation channel $\boldsymbol{P}(\theta) := \rho \mapsto e^{-i\theta/2P}\rho e^{i\theta/2P}$, where P is a Pauli operation (with plus or minus sign) and $0 \le \theta \le \pi$. We can find a Clifford unitary U that maps P to Z_1. Then $\boldsymbol{P}(\theta)$ can be decomposed as:

$$\begin{aligned} \boldsymbol{P}(\theta) = \rho \mapsto\ & U^\dagger e^{-i\theta/2Z_1} U\rho U^\dagger e^{i\theta/2Z_1} U \\ &= \boldsymbol{U}^\dagger \cdot \boldsymbol{Z}_1(\theta) \cdot \boldsymbol{U} \\ &= \frac{1 + \cos\theta - \sin\theta}{2}\boldsymbol{I} + \frac{1 - \cos\theta - \sin\theta}{2}\boldsymbol{U}^\dagger\boldsymbol{Z}_1\boldsymbol{U} + \sin\theta\boldsymbol{U}^\dagger\boldsymbol{S}\boldsymbol{U} \end{aligned}$$

which is also nearly optimal for $0 \le \theta \le \pi$. Using the Clifford channel decomposition, we can directly apply these three Clifford channels on SPD separately and then merge the resulted SPDs.

5 EVALUATION

5.1 Experimental Settings

The algorithms proposed in the previous sections are implemented on Python3, with the help of IBM Qiskit [1] — an open source SDK for quantum computation — and CVXPY [11] — an open source Python-embedded modeling language for convex optimization problems. Note that we use the Clifford circuit synthesis algorithm implemented by Qiskit, which is based on the method in [8] and yields lower circuit size (though non-optimal in general).

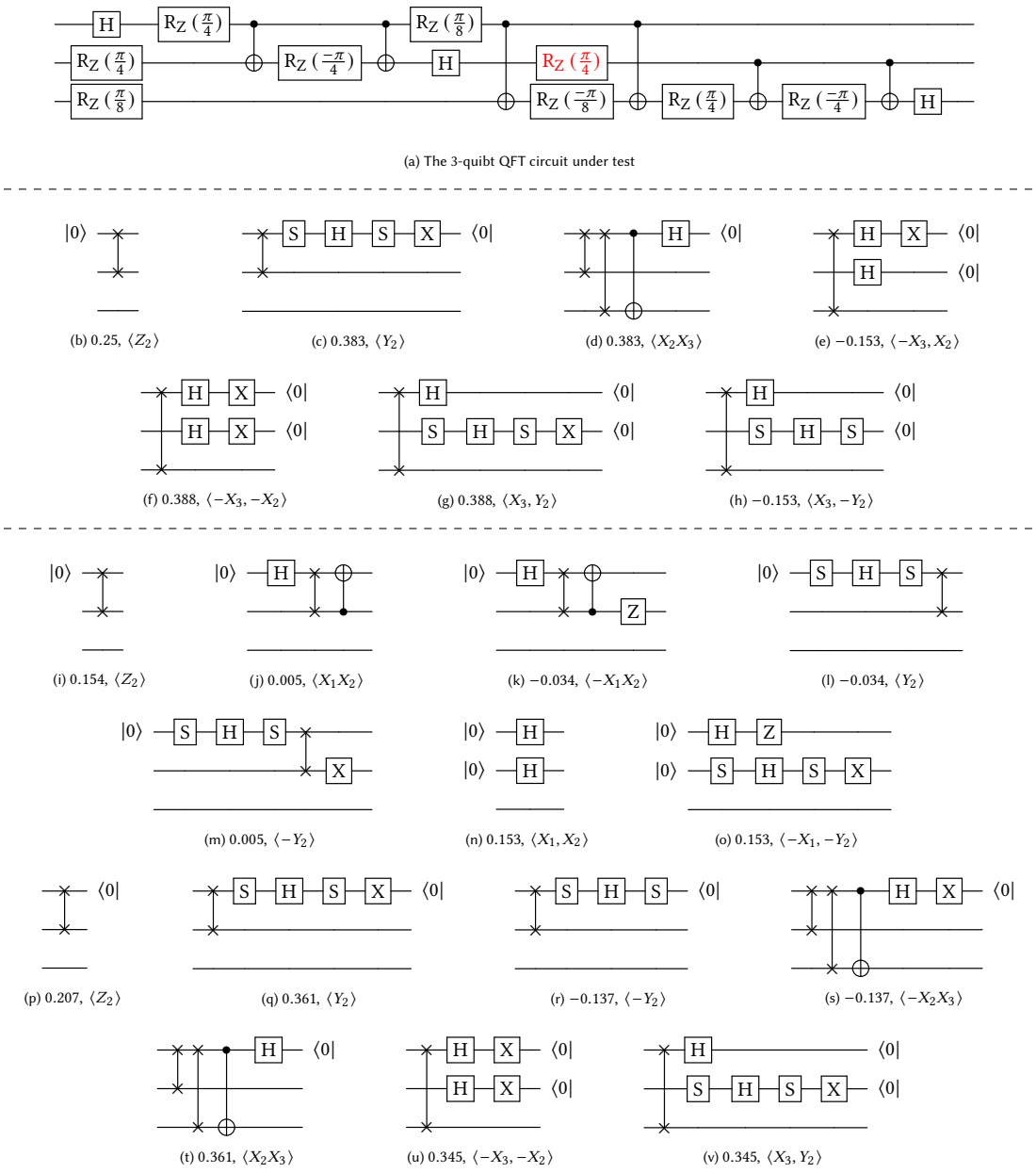
To demonstrate the effectiveness and scalability of our method, we conduct the experiments on several commonly used quantum circuits of different sizes, including Quantum Fourier Transform (QFT) [33], Bernstein-Vazirani (BV) [6] and Quantum Volume (QV) [32]. The Quantum Fourier Transform is the quantum analogue of the discrete Fourier transform, which is used in many quantum algorithms such as Shor's factoring, quantum hidden subgroup algorithm, as well as the QFT-based adder. The Bernstein-Vazirani algorithm is a quantum algorithm that can efficiently learn the secret string s of a linear Boolean function $f(x) = x \cdot s$. The Quantum Volume circuits are used to quantify the ability (such as the operation fidelity, connectivity and available gate sets) of near-term quantum computing systems. The detailed information such as circuit size, the depth and number of non-Clifford gates of our benchmark circuits is summarized in Table 1.

Table 1. Benchmark Circuits

	Qubits	Size	Depth	Non-Clifford Gates
QFT_3	3	18	14	9
QFT_5	5	55	30	30
QFT_10	10	235	70	135
QV_5	5	135	43	105
QV_7	7	201	43	156
BV_10	10	29	12	0
BV_100	100	299	102	0

In our experiments, we adopt the missing-gate fault model [5, 23, 34], where the gate is assumed to be missing (*i.e.*, the gate is replaced by the identity operation). However, any unitary fault is equivalently handled by our algorithms, with no additional overhead in general. An example is shown in Fig. 6, where both the missing-gate fault and a general unitary fault (the replaced-by-$R_X(\pi/3)$ fault in this example) are handled on QFT_3 circuit.

The remaining part of this section is arranged as follows. In subsection 5.2, we evaluate the proposed SPD-generation algorithm (Algorithm 2) for test pattern generation, on several benchmark circuits. In subsection 5.3, we evaluate the proposed sampling algorithm (Algorithm 1) for test application, using the previously generated test pattern, and the overall performance of our method for the fault detection of quantum circuit is also reported.



(a) The 3-quibt QFT circuit under test

(b) 0.25, $\langle Z_2 \rangle$ (c) 0.383, $\langle Y_2 \rangle$ (d) 0.383, $\langle X_2X_3 \rangle$ (e) -0.153, $\langle -X_3, X_2 \rangle$

(f) 0.388, $\langle -X_3, -X_2 \rangle$ (g) 0.388, $\langle X_3, Y_2 \rangle$ (h) -0.153, $\langle X_3, -Y_2 \rangle$

(i) 0.154, $\langle Z_2 \rangle$ (j) 0.005, $\langle X_1X_2 \rangle$ (k) -0.034, $\langle -X_1X_2 \rangle$ (l) -0.034, $\langle Y_2 \rangle$

(m) 0.005, $\langle -Y_2 \rangle$ (n) 0.153, $\langle X_1, X_2 \rangle$ (o) 0.153, $\langle -X_1, -Y_2 \rangle$

(p) 0.207, $\langle Z_2 \rangle$ (q) 0.361, $\langle Y_2 \rangle$ (r) -0.137, $\langle -Y_2 \rangle$ (s) -0.137, $\langle -X_2X_3 \rangle$

(t) 0.361, $\langle X_2X_3 \rangle$ (u) 0.345, $\langle -X_3, -X_2 \rangle$ (v) 0.345, $\langle X_3, Y_2 \rangle$

Fig. 6. An illustrative example of the quantum test patterns represented by SPD. Sub-figure (a) is the 3-qubit QFT circuit using Clifford + $\exp(i\theta P)$ gates. The red colored $R_Z(\pi/4)$ gate is suspected to be faulty, and the following test patterns are automatically generated on two different fault models: **1)** Sub-figures (b)-(h) represent the test patterns for detecting the missing-gate fault, where sub-figure (b) is the input pattern and sub-figures (c)-(h) are the measurement patterns. **2)** Sub-figures (i)-(v) represent the test patterns for detecting a more general unitary fault, *i.e.*, replaced-by-$R_X(\pi/3)$ fault, where sub-figures (i)-(o) are the input patterns and sub-figures (p)-(v) are the measurement patterns. The number and the Pauli operators in the sub-caption of each circuit represent the coefficient and corresponding stabilizer group in the SPD.

5.2 Test Generation

In this section, we apply the proposed SPD generation algorithm on different benchmark circuits and evaluate the generated test patterns. First, an illustrative example is shown in Fig. 6. In this example, the SPD generation algorithm is applied on the 3-qubit QFT circuit (see sub-figure (a)), where the middle red-colored $R_z(\pi/4)$ gate is suspected to be faulty. Two different fault models are adopted and the corresponding test patterns are generated respectively. The sub-figures (b)-(h) are test patterns for the missing-gate fault, and the sub-figures (i)-(v) are test patterns for the replaced-by-$R_X(\pi/3)$ fault. The number and signed Pauli operators in the caption of each sub-figures (excluding sub-figure (a)) represent the coefficient and corresponding stabilizer projector of the SPD. The test patterns that have input (output) labels are input (output) patterns. The input labels represent the specified input, and those qubits without input labels are in the maximally mixed state. The output labels represent the specified measurement output, and those qubits without output labels are simply discarded.

We then evaluate the performance of the proposed SPD generation algorithm in detail. Specifically, we are concerned with the following metrics: **1)** the product of the SPD 1-norm $\nu^*(\mathcal{A}_\rho)\nu(\mathcal{B}_M)$, where $\mathcal{A}_\rho$ denotes the generated SPD for test input ρ and $\mathcal{B}_M$ denotes the generated SPD for test measurement M, **2)** the sparsity of the generated SPD, **3)** the size of the quantum Clifford circuit that implements the test pattern, and **4)** the depth of the quantum Clifford circuit that implements the test pattern. The first metric $\nu^*(\mathcal{A}_\rho)\nu(\mathcal{B}_M)$ could be treated as a quantity that measures how "non-Clifford" the SPD is. It heavily impacts the iteration number needed in the sampling algorithm (see Algorithm 1). The second metric reflects the computational complexity required for any non-trivial operation on the SPD. The third and fourth metrics reflect the complexity of the Clifford circuit implementing the test patterns.

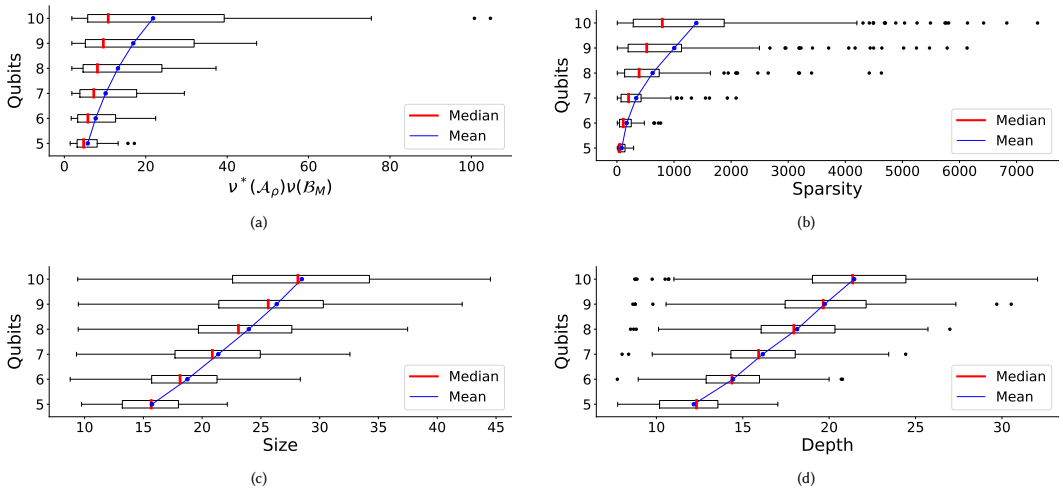


Fig. 7. Different metrics for the generated test patterns. The evaluation is based on QFT circuits and the circuit sizes range from 5 qubits to 10 qubits. The box extends from the first quartile to the third quartile of the data.

First, we conduct the experiments on QFT circuits with the sizes ranging from 5 qubits to 10 qubits. For each potential fault in the circuit, the corresponding test patterns (represented by SPD) are generated by our algorithm, and then evaluated with the aforementioned metrics. The results are shown in Fig. 7. It is worth noting that, though

Table 2. Averaged metric values of the generated test patterns on different benchmark circuits. The numbers in parentheses represent the optimal circuit size (depth) if we carefully choose the initial subspace Q (see Equation (8)).

	$\nu^*(\mathcal{A}_\rho)$	$\nu(\mathcal{B}_M)$	Sparsity	Size	Depth
QFT_5	1.698	3.381	84.0	15.7	12.2
QFT_6	1.956	4.056	171.5	18.7	14.4
QFT_7	2.272	4.764	338.4	21.4	16.2
QFT_8	2.653	5.484	624.8	24.0	18.1
QFT_9	3.112	6.222	1004.7	26.3	19.7
QFT_10	3.736	6.938	1390.8	28.5	21.4
QV_5	2.119	8.632	835.0	26.8	19.2
QV_7	3.209	15.006	5118.4	33.3	22.8
BV_10	1.493	1.479	10.3	18.9	15.9
BV_100	1.513	1.497	10.6	162.0 (37)	145.2 (7)

the maximum of $\nu^*(\mathcal{A}_\rho)\nu(\mathcal{B}_M)$ grows rapidly, the median of $\nu^*(\mathcal{A}_\rho)\nu(\mathcal{B}_M)$ grows slowly with the number of qubits. This demonstrates good scalability of the SPD in a considerable proportion of the cases. Similarly, the median of sparsity scales moderately compared to its maximum value, which demonstrates the computational efficiency of the SPD generation in the average sense. We believe that the SPD generation algorithm can be further accelerated using advanced parallel techniques on classical computer. We also observe that the averaged circuit size and depth of the test pattern scale almost linearly, which further ensures the practical feasibility of the proposed test application scheme. Then, we also evaluate the QV and BV circuits, where the results are shown in Table 2. Note that for the 7-qubit QV circuit, the metrics $\nu^*(\mathcal{A}_\rho)$, $\nu(\mathcal{B}_M)$ and the sparsity are rather higher than others. This is likely because QV is a kind of randomized quantum circuits designed for quantifying the maximal capacity of a quantum device, so that it is much more complicated than most of the quantum circuits for normal uses. In contrast, the BV circuit contains only Clifford gates, so that our algorithm even scales well for the circuit size up to 100 qubits.

5.3 Test Application and Fault Detection

In this section, we first justify the prior mentioned quantum-classical resources trade off (see Fig. 3 and related discussion). To see this, we compare the "direct" method with our SPD-based method, as shown in Table 3. Both methods implement the same test patterns (i.e., (ρ, M)), where the "direct" method uses Clifford + Pauli-rotation gate set, while the SPD-based method uses only Clifford gate set. Specifically, we conduct the experiments on QFT, QV and BV circuits with different sizes. For each potential fault in the CUT, the corresponding test pattern is generated and evaluated, and the averaged results are given in Table 3. Note that the sizes and depths of the circuits implementing the test patterns by our SPD-based method are significantly less than those by "direct" method. Moreover, our method only uses Clifford circuits. These demonstrate its efficiency and robustness.

Then, we apply the test patterns (generated in Section 5.2) on benchmark circuits by simulating the sampling algorithm (Algorithm 1). Then, we show the overall results of our framework for quantum circuit fault detection. First, an illustrative example is shown in Fig. 8. In this example, we simulate the sampling algorithm on the 3-qubit fault-free

Table 3. Comparison between different test application methods. "Direct" method uses the Clifford + Pauli-rotation gate set to implement the test patterns, while SPD-based method uses only the Clifford gate set. We report the averaged circuit sizes and depths, which reflects the test equipment complexity, and also the averaged $\nu^*(\mathcal{A})\nu(\mathcal{B})$, which reflects the overhead in test time complexity.

	Direct		SPD-Based		
	Size	Depth	Size	Depth	$\nu^*(\mathcal{A})\nu(\mathcal{B})$
QFT_9	200.7	76.9	26.3	19.7	19.4
QFT_10	246.8	86.9	28.5	21.4	25.9
QV_5	141.6	54.6	26.8	19.2	18.3
QV_7	207.7	59.0	33.3	22.8	48.2
BV_10	36.8	16.7	18.9	15.9	2.2
BV_100	307.3	107.0	162.0	145.2	2.3

QFT circuit, using the test pattern for each gate in the circuit. The orange and green bars represent the simulation results with different parameters, and the blue bar represents its expected value. The gate ID i indicates that the result is from applying the test patterns of i-th gate. First note that when setting $\delta = 0.1, \epsilon = 0.1$, the estimation returned by the sampling algorithm successfully approximates the ideal output. We found that when applying the test patterns of gate 6 and gate 7, the green bar is lower than 0.5. This means we may misclassify the CUT as faulty circuit when we set the classification threshold to be 0.5, resulting in a false positive alert. However, this is mainly because the gate itself is close to the identity gate to some extent. Thus it is intrinsically difficult to detect such fault under the SMGF (single missing gate fault) model. Except gate-6 and gate-7, there is no big difference between the results of using parameters $\delta = 0.1, \epsilon = 0.1$ and $\delta = 0.3, \epsilon = 0.3$ while the latter one requires fewer iterations. This means if the faults are sufficiently distinguishable, then the parameters $\delta = 0.3, \epsilon = 0.3$ may be a practical choice that are both efficient and sound.

Then we evaluate our algorithms for the fault detection task. We first randomly selected k candidate fault-sites in the benchmark circuit such that the maximal discrimination probability is at least $0.5 + \tau$. This is because the discrimination probability of $0.5 + \tau$ indicates that the trace distance between fault-free and faulty output is at most 2τ. Thus $\tau \to 0$ indicates that the fault is negligible and inherently hard to detect. With the probability of 0.5, the CUT is set to be faulty, and otherwise the CUT is set to be fault-free. When the CUT is faulty, a fault-site is randomly selected from the k candidates and the gate at the fault-site is replaced by its faulty version. For each candidate fault-site i, the corresponding test patterns are automatically generated and applied on the CUT by simulating the sampling algorithm (with parameters ϵ, δ) to obtain the estimation m_i. The CUT is predicted to be faulty if $\min\{m_1, \ldots, m_k\} \leq 0.5$, and otherwise is predicted to be fault-free. Note that we need to correctly obtain k estimations to ensure that the final prediction is correct. Since the parameter ϵ bounds the success probability of our sampling algorithm from below, we set this parameter to ϵ/k to ensure that the overall success probability for fault detection is at least $1 - \epsilon$. Note that the complexity of our sampling algorithm scales logarithmically with $1/\epsilon$, thus the overhead is still acceptable. In this work, we set $k = 10, \tau = 0.1, \delta = 0.3, \epsilon = 0.3$. For each benchmark circuit, we repeat the above fault detection procedure 100 times and the experimental results are shown in Table 4.

Our method achieves the highest recall (1.0) uniformly in all experiments, which means all the faults in the quantum circuits are detected with success probability 1.0. We also found that there are some false positive alerts in the experiments for QFT_5 and QFT_10 circuits, *i.e.*, some fault-free circuits are misclassified as faulty circuits. However, the precision is

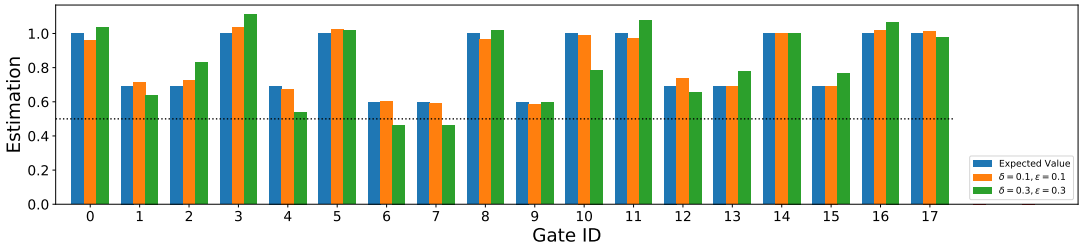


Fig. 8. An illustrative example for the sampling algorithm (Algorithm 1) on 3-qubit QFT circuit. The orange and green bars represent the simulation results with different parameters, and the blue bar represents the expected output. The gate ID i means that we apply the test patterns for the i-th gate.

Table 4. Simulation results for quantum fault detection.

	TP	TN	FP	FN	Prec.	Rec.	Acc.
QV_5	51	49	0	0	1.0	1.0	1.0
BV_10	47	53	0	0	1.0	1.0	1.0
QFT_5	46	47	7	0	0.87	1.0	0.93
QFT_10	44	47	9	0	0.83	1.0	0.91

still higher than 0.83 and the false positive rate (FPR) is lower than 0.16. Furthermore, since the recall is sufficiently close to 1.0, the false positive alerts can be effectively eliminated by repeating the test on the false positive samples and excluding the samples once they are classified as negative (*i.e.*, fault-free). Finally, the overall performance of fault detection is much better than expected by the theoretical bounds (failure probability $\epsilon = 0.3$ and estimation deviation $\delta = 0.3$), which indicates rather good performance of our method in practice.

6 CONCLUSION

In this paper, we propose a new ATPG framework for robust quantum circuit testing. The stabilizer projector decomposition (SPD) is introduced for representing the quantum test pattern, and an SPD generation algorithm is developed. Furthermore, the test application is reconstructed using Clifford-only circuits with classical randomness. To tackle the issue of exponentially growing size of the involved optimization problem, we proposed several acceleration techniques that can exploit both locality and sparsity in SPD generation. It is proved that the locality exploiting algorithm will return optimal result under some reasonable conditions. To demonstrate the effectiveness and efficiency, the overall framework is implemented and evaluated on several commonly used benchmark circuits.

For future research, we plan to extend our work in the following two directions:

6.0.1 Adaptive quantum ATPG. The adaptive ATPG method [26] for classical circuits utilizes the existing output responses from the CUT to guide the automatic generation of new test patterns, which makes the fault diagnosis much more efficient. In the quantum case, however, the output responses are quantum states, and thus we cannot easily make use of the information in the output responses due to the high cost of quantum state tomography. Further, the response

analysis and pattern generation guidance are much more complicated than those for classical circuit. These make the development of adaptive quantum ATPG high nontrivial.

6.0.2 Different Fault Models and Circuit Models. In this paper, we assume that the faulty gate is still a unitary gate. This assumption does cover some important classes of faults in the real-world applications. But in many applications, a fault gate is often a non-unitary gate. Then techniques for discriminating a unitary gate and a general quantum operation (see e.g. [15]) is vital in ATPG for this kind of faults. On the other hand, we only consider combinational quantum circuits in this paper. Recently, several new circuit models have been introduced in quantum computing and quantum information. One interesting model is dynamic quantum circuits (see e.g. [9, 10]) in which quantum measurements can occur at the middle of the circuits and the measurement outcomes are used to conditionally control subsequent steps of the computation. A fault in these circuits may happen not only at a quantum gate but also at a measurement. Therefore, techniques for discriminating quantum measurements (see e.g. [28]) will be needed in ATPG for them. Another interesting model is sequential quantum circuits [43]. As we know, in the classical case, ATPG for combinational circuits and that for sequential ones are fundamentally different. The quantum case is similar, and the approach presented in this paper cannot directly generalised to sequential quantum circuits.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61832015.

REFERENCES

[1] [n. d.]. Qiskit. https://qiskit.org.
[2] Scott Aaronson and Daniel Gottesman. 2004. Improved simulation of stabilizer circuits. *Physical Review A* 70, 5 (2004), 052328.
[3] Antonio Acin. 2001. Statistical distinguishability between unitary operations. *Physical review letters* 87, 17 (2001), 177901.
[4] Ryan S Bennink, Erik M Ferragut, Travis S Humble, Jason A Laska, James J Nutaro, Mark G Pleszkoch, and Raphael C Pooser. 2017. Unbiased simulation of near-Clifford quantum circuits. *Physical Review A* 95, 6 (2017), 062337.
[5] Debajyoti Bera. 2017. Detection and diagnosis of single faults in quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 3 (2017), 587–600.
[6] Ethan Bernstein and Umesh Vazirani. 1997. Quantum complexity theory. *SIAM Journal on computing* 26, 5 (1997), 1411–1473.
[7] Sergey Bravyi and Alexei Kitaev. 2005. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A* 71, 2 (2005), 022316.
[8] Sergey Bravyi, Ruslan Shaydulin, Shaohan Hu, and Dmitri Maslov. 2021. Clifford circuit optimization with templates and symbolic pauli gates. *arXiv preprint arXiv:2105.02291* (2021).
[9] Lukas Burgholzer and Robert Wille. 2021. Towards verification of dynamic quantum circuits. *arXiv preprint arXiv:2106.01099* (2021).
[10] Antonio D Córcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K Minev, Jerry M Chow, and Jay M Gambetta. 2021. Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits. *Physical Review Letters* 127, 10 (2021), 100501.
[11] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.
[12] Joseph Emerson, Robert Alicki, and Karol Życzkowski. 2005. Scalable noise estimation with random unitary operators. *Journal of Optics B: Quantum and Semiclassical Optics* 7, 10 (2005), S347.
[13] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. 2012. Surface codes: Towards practical large-scale quantum computation. *Physical Review A* 86, 3 (2012), 032324.
[14] Austin G Fowler, Ashley M Stephens, and Peter Groszkowski. 2009. High-threshold universal quantum computation on the surface code. *Physical Review A* 80, 5 (2009), 052312.
[15] Alexei Gilchrist, Nathan K Langford, and Michael A Nielsen. 2005. Distance measures to compare real and ideal quantum processes. *Physical Review A* 71, 6 (2005), 062310.
[16] Daniel Gottesman. 1998. The Heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006* (1998).
[17] David Gross. 2006. Hudson's theorem for finite-dimensional quantum systems. *Journal of mathematical physics* 47, 12 (2006), 122107.
[18] Lov K Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 212–219.

[19] Robin Harper and Steven T Flammia. 2019. Fault-tolerant logical gates in the IBM quantum experience. *Physical review letters* 122, 8 (2019), 080504.
[20] Robin Harper, Steven T Flammia, and Joel J Wallman. 2020. Efficient learning of quantum noise. *Nature Physics* 16, 12 (2020), 1184–1188.
[21] Nicholas C. Harris, Yangjin Ma, Jacob Mower, Tom Baehr-Jones, Dirk Englund, Michael Hochberg, and Christophe Galland. 2014. Efficient, compact and low loss thermo-optic phase shifter in silicon. *Optics express* 22, 9 (2014).
[22] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum algorithm for linear systems of equations. *Physical review letters* 103, 15 (2009), 150502.
[23] John P Hayes, Ilia Polian, and Bernd Becker. 2004. Testing for missing-gate faults in reversible circuits. In *13th Asian test symposium*. IEEE, 100–105.
[24] Carl W Helstrom. 1969. Quantum detection and estimation theory. *Journal of Statistical Physics* 1, 2 (1969), 231–252.
[25] Alexander S Holevo. 1973. Statistical decision theory for quantum systems. *Journal of multivariate analysis* 3, 4 (1973), 337–394.
[26] Stefan Holst and Hans-Joachim Wunderlich. 2009. Adaptive debug and diagnosis without fault dictionaries. *Journal of Electronic Testing* 25, 4 (2009), 259–268.
[27] Mark Howard and Earl Campbell. 2017. Application of a resource theory for magic states to fault-tolerant quantum computing. *Physical review letters* 118, 9 (2017), 090501.
[28] Zhengfeng Ji, Yuan Feng, Runyao Duan, and Mingsheng Ying. 2006. Identification and distance measures of measurement apparatus. *Physical Review Letters* 96, 20 (2006), 200401.
[29] Miguel Sousa Lobo, Maryam Fazel, and Stephen Boyd. 2007. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research* 152, 1 (2007), 341–365.
[30] Easwar Magesan, Jay M Gambetta, and Joseph Emerson. 2011. Scalable and robust randomized benchmarking of quantum processes. *Physical review letters* 106, 18 (2011), 180504.
[31] Olvi L Mangasarian and RR Meyer. 1979. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization* 17, 6 (1979), 745–752.
[32] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli, and Kristan Temme. 2018. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology* 3, 3 (2018), 030503.
[33] Michael A Nielsen and Isaac L Chuang. 2010. *Quantum computation and quantum information*. Cambridge university press.
[34] Alexandru Paler, Ilia Polian, and John P Hayes. 2012. Detection and diagnosis of faulty quantum circuits. In *17th Asia and South Pacific Design Automation Conference*. IEEE, 181–186.
[35] Hakop Pashayan, Joel J Wallman, and Stephen D Bartlett. 2015. Estimating outcome probabilities of quantum circuits using quasiprobabilities. *Physical review letters* 115, 7 (2015), 070501.
[36] Emanuele Pelucchi, Giorgos Fagas, Igor Aharonovich, Dirk Englund, Eden Figueroa, Qihuang Gong, Hübel Hannes, Jin Liu, Chao-Yang Lu, Nobuyuki Matsuda, Jian-Wei Pan, Florian Schreck, Fabio Sciarrino, Christine Silberhorn, Jianwei Wang, and K. Jöns. 2022. The potential and global outlook of integrated photonics for quantum technologies. *Nature Reviews Physics* 4, 3 (2022), 194–208.
[37] Lukas Postler, Sascha Heuβen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth, Christian D. Marciniak, Roman Stricker, Martin Ringbauer, Rainer Blatt, Philipp Schindler, Markus Müller, and Thomas Monz. 2022. Demonstration of fault-tolerant universal quantum gate operations. *Nature* 605, 7911 (2022), 675–680.
[38] Hammam Qassim, Joel J Wallman, and Joseph Emerson. 2019. Clifford recompilation for faster classical simulation of quantum circuits. *Quantum* 3 (2019), 170.
[39] J Paul Roth. 1966. Diagnosis of automata failures: A calculus and a method. *IBM journal of Research and Development* 10, 4 (1966), 278–291.
[40] Peter W Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 124–134.
[41] Jianwei Wang, Fabio Sciarrino, Anthony Laing, and Mark G. Thompson. 2020. Integrated photonic quantum technologies. *Nature Photonics* 14, 5 (2020), 273–284.
[42] Laung-Terng Wang, Cheng-Wen Wu, and Xiaoqing Wen. 2006. *VLSI test principles and architectures: design for testability*. Elsevier.
[43] Qisheng Wang, Riling Li, and Mingsheng Ying. 2021. Equivalence checking of sequential quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021).
[44] Xinlan Zhou, Debbie W Leung, and Isaac L Chuang. 2000. Methodology for quantum logic gate construction. *Physical Review A* 62, 5 (2000), 052316.

A APPENDIX

In this appendix, we provide the detailed proofs of all lemmas, propositions and theorems omitted in the main body of the paper.

A.1 Minimal Norms of SPD

Definition 3 (Minimal SPD norms). *The minimal 1-norm ξ and minimal weighted 1-norm ξ^* of operator A are defined by*

$$\begin{aligned} \xi(A) &= \min\left\{ \nu(\mathcal{A}) \;\middle|\; \mathcal{A} \in \mathrm{SPD}(A) \right\} \\ \xi^*(A) &= \min\left\{ \nu^*(\mathcal{A}) \;\middle|\; \mathcal{A} \in \mathrm{SPD}(A) \right\} \end{aligned} \tag{26}$$

We say that $\mathcal{A}$ is ν-optimal (respectively, ν^*-optimal) for A if $\mathcal{A} \in \mathrm{SPD}(A)$ and $\nu(\mathcal{A}) = \xi(A)$ (respectively, $\nu^*(\mathcal{A}) = \xi^*(A)$).

The above definition is a generalization of the robustness of magic in quantum resource theory [27]. Some basic properties of ξ and ξ^* are given in the following:

Proposition 2. *Both ξ and ξ^* are invariant under applying Clifford unitary and tensoring stabilizer projector:*

(1) *Clifford Invariance: for any Clifford unitary U,*

$$\xi(UAU^\dagger) = \xi(A), \quad \xi^*(UAU^\dagger) = \xi^*(A).$$

(2) *Tensor Invariance: for any stabilizer projector A,*

$$\xi(A \otimes B) = \xi(B), \quad \xi^*(A \otimes B) = \mathrm{tr}A \cdot \xi^*(B).$$

The Clifford invariance is obvious and the proof of tensor invariance is given in the next section.

A.2 Proof of Proposition 2 (Tensor Invariance)

The proof of Proposition 2 requires several technical lemmas:

Lemma 2. *Suppose A is a stabilizer projector of bipartite system ST, where S is a single qubit subsystem. Then*

$$\langle 0|A|0\rangle = cB \tag{27}$$

where B is a stabilizer projector of T and $0 \le c \le 1$. Furthermore, if $c = 1$, then $A \sqsupseteq |0\rangle\langle 0| \otimes B$.

Proof. Suppose $G = \langle P_1, \ldots, P_l \rangle$ is the signed Pauli group associated to the stabilizer projector A where $P_1, \ldots, P_l$ are independent generators of G. Without loss of generality, we can assume that either Z_1 commutes with all P_i or only anti-commutes with P_l.

If Z_1 commutes with all P_i, then

$$\begin{aligned} |0\rangle\langle 0| \otimes cB &= |0\rangle\langle 0|A|0\rangle\langle 0| \\ &= \frac{I+Z_1}{2} A \frac{I+Z_1}{2} \\ &= \frac{I+Z_1}{2} \prod_{i \le l} \frac{I+P_i}{2} \frac{I+Z_1}{2} \\ &= \prod_{i \le l} \frac{I+P_i}{2} \frac{I+Z_1}{2} \end{aligned} \tag{28}$$

Since each P_i commutes with Z_1, $\langle P_1, \ldots, P_l, Z_1\rangle$ is a stabilizer group. Thus we can replace P_i by $P_i Z_1$ (if necessary) to ensure that P_i is of the form $I \otimes P_i'$. This means

$$|0\rangle\langle 0| \otimes cB = |0\rangle\langle 0| \otimes \prod_{i\le l} \frac{I+P_i'}{2} \tag{29}$$

Thus $c = 1$ and B is a stabilizer projector.

If Z_1 anti-commutes with P_l, then

$$\begin{aligned} |0\rangle\langle 0| \otimes cB &= |0\rangle\langle 0| A |0\rangle\langle 0| \\ &= \frac{I+Z_1}{2} A \frac{I+Z_1}{2} \\ &= \prod_{i\le l-1} \frac{I+P_i}{2}\frac{I+Z_1}{2}\frac{I+P_l}{2}\frac{I+Z_1}{2} \\ &= \frac{1}{2}\prod_{i\le l-1} \frac{I+P_i}{2}\frac{I+Z_1}{2} \end{aligned} \tag{30}$$

Similarly, we can assume P_i is of the form $I \otimes P_i'$, which means

$$|0\rangle\langle 0| \otimes cB = |0\rangle\langle 0| \otimes \frac{1}{2}\prod_{i\le l-1} \frac{I+P_i'}{2} \tag{31}$$

Thus $c = 1/2$ and B is a stabilizer projector.

Furthermore, $c = 1$ indicates that all P_i commutes with Z_1. So each P_i is either of the form $I \otimes P_i'$ or of the form $Z \otimes P_i'$. If $P_i = I \otimes P_i'$ for all i, then

$$A = I \otimes \prod \frac{I+P_i'}{2} = I \otimes B \sqsupseteq |0\rangle\langle 0| \otimes B \tag{32}$$

Otherwise, there is some generator $P_i = Z \otimes P_i'$. Without loss of generality, we can assume that only P_l has the form $Z \otimes P_l'$. Then

$$\begin{aligned} A &= (I \otimes \prod_{i<l} \frac{I+P_i'}{2}) \frac{I + Z\otimes P_l'}{2} \\ &= (I \otimes \prod_{i<l} \frac{I+P_i'}{2}) \big(|0\rangle\langle 0| \otimes \frac{I+P_l'}{2} + |1\rangle\langle 1| \otimes \frac{I-P_l'}{2}\big) \\ &= |0\rangle\langle 0| \otimes \prod_{i\le l} \frac{I+P_i'}{2} + |1\rangle\langle 1| \otimes \prod_{i<l} \frac{I+P_i'}{2}\frac{I-P_l'}{2} \\ &\sqsupseteq |0\rangle\langle 0| \otimes B \end{aligned}$$

□

Lemma 3. *Suppose A is a stabilizer projector of bipartite system ST, where S is an n-qubit system, then*

$$\langle 0|^{\otimes n} A |0\rangle^{\otimes n} = cB \tag{33}$$

where B is a stabilizer projector of T and $0 \le c \le 1$. Furthermore, if $c = 1$, then $A \sqsupseteq |0\rangle\langle 0|^{\otimes n} \otimes B$

Proof. By inductively apply Lemma 2. □

Lemma 4. *Suppose A is a stabilizer projector of bipartite system ST, then*

$$\mathrm{tr}_S(A) = cB \tag{34}$$

where B is a stabilizer projector and $0 \leq c \leq \dim(S)$. Furthermore, if $c = \dim(S)$, then $A = I \otimes B$.

Proof. Suppose $G = \{P_i\}_i$ is the stabilizer group associated to the stabilizer projector A. Obviously, all the signed Pauli operators P_i that act trivially on system T (*i.e.*, $P_i = I \otimes P_i'$) form a subgroup G'. Then,

$$\begin{aligned}
\mathrm{tr}_S(A) &= \frac{1}{|G|} \sum_{P_i \in G} \mathrm{tr}_S(P_i) \\
&= \frac{1}{|G|} \sum_{P_i \in G'} \mathrm{tr}_S(P_i) \\
&= \frac{\dim(S)}{|G|} \sum_{P_i \in G'} P_i' \\
&= \frac{\dim(S)|G'|}{|G|} \frac{1}{|G'|} \sum_{P_i \in G'} P_i' \\
&= \frac{\dim(S)|G'|}{|G|} B
\end{aligned} \tag{35}$$

So $c = \dim(S)|G'|/|G| \leq \dim(S)$ and B is a stabilizer projector. Furthermore, if $c = \dim(S)$, then $|G'| = |G|$. This means $G' = G$ and thus $P_i = I \otimes P_i'$ for all $P_i \in G$. So $A = I \otimes B$. □

Now we are ready to give the proof of Proposition 2. For the clarity, let us split the proposition into the following two:

Proposition 3 (Tensor Invariance of ξ).

$$\xi(A \otimes B) = \xi(B) \tag{36}$$

where A is a stabilizer projector.

Proof. We use S and T to denote the subsystems corresponding to the operators A and B respectively. By Corollary 1, there is a Clifford unitary U such that $UAU^\dagger = |0\rangle\langle 0|^{\otimes m} \otimes I_n$, where $m = \log \dim(S) - \log \mathrm{tr}(A)$ and $n = \log \mathrm{tr}(A)$. Suppose $\{(a_i, A_i)\}$ is an optimal SPD for $|0\rangle\langle 0|^{\otimes m} \otimes I_n \otimes B$, so

$$|0\rangle\langle 0|^{\otimes m} \otimes I_n \otimes B = \sum_i a_i A_i \tag{37}$$

By Lemma 3

$$I_n \otimes B = \sum_i a_i \langle 0|^{\otimes m} A_i |0\rangle^{\otimes m} = c_1 \sum_i a_i A_i' \tag{38}$$

where $0 \leq c_1 \leq 1$ and A_i' are stabilizer projectors. By Lemma 4

$$2^n \cdot B = c_1 \sum_i a_i \mathrm{tr}_n(A_i') = c_1 c_2 \sum_i a_i A_i'' \tag{39}$$

where $0 \leq c_2 \leq 2^n$, tr_n denotes the partial trace on first n quibts and A_i'' are stabilizer projectors. Thus

$$B = c_1 \cdot c_2 / 2^n \cdot \sum_i a_i A_i'' \tag{40}$$

This implies $\xi(B) \leq \xi(|0\rangle\langle 0|^{\otimes m} \otimes I_n \otimes B) = \xi(A \otimes B)$. But obviously we have $\xi(A \otimes B) \leq \xi(B)$. Thus $\xi(A \otimes B) = \xi(B)$. □

Proposition 4 (Tensor Invariance of ξ^*).

$$\xi^*(A \otimes B) = \mathrm{tr}A \cdot \xi^*(B) \tag{41}$$

where A is a stabilizer projector.

Proof. Since any stabilizer projector S can be expanded into the sum of $\mathrm{tr}S$ stabilizer states, we can show that there is an stabilizer state decomposition

$$A \otimes B = \sum_i a_i \sigma_i \tag{42}$$

where σ_i are stabilizer states and $\sum_i |a_i| = \xi^*(A \otimes B)$. We use S to denote the subsystem of operator A. By Lemma 4

$$\mathrm{tr}A \cdot B = \sum_i a_i \mathrm{tr}_S(\sigma_i) = \sum_i a_i c_i \sigma_i' \tag{43}$$

where σ_i' are stabilizer projectors and $c_i \mathrm{tr}(\sigma_i') = \mathrm{tr}(\sigma_i) = 1$. This means

$$B = \sum_i a_i \frac{1}{\mathrm{tr}A \cdot \mathrm{tr}(\sigma_i')} \sigma_i' \tag{44}$$

This implies $\xi^*(B) \leq 1/\mathrm{tr}A \cdot \sum_i |a_i| = 1/\mathrm{tr}A \cdot \xi^*(A \otimes B)$. But obviously we have $\xi^*(A \otimes B) \leq \mathrm{tr}(A)\xi^*(B)$. Thus $\xi^*(A \otimes B) = \mathrm{tr}A \cdot \xi^*(B)$ □

A.3 Proof of Theorem 2 (Optimality of Algorithm 3)

First, we show that the SPD is in a canonical form if one of the two conditions in Theorem 2 is satisfied. For clarity, we will prove the two cases separately.

Proposition 5. *Suppose A is a stabilizer projector and B is a positive operator. If $C = \{(c_i, C_i)\} \in \mathrm{SPD}(A \otimes B)$ satisfies*

(1) *C is v-optimal,*
(2) *C has the smallest v^* among all v-optimal SPDs of $A \otimes B$*

then each C_i has the form $C_i = A \otimes C_i'$

Proof. Suppose A is an n-qubit stabilizer projector and let $m = \log \mathrm{tr}A$. Then, there is a Clifford unitary U such that $UAU^\dagger = |0\rangle\langle 0|^{\otimes n-m} \otimes I_m$, where I_m is the identity operator on the m-qubit system. Note that U is Clifford, if $\{(c_i, C_i)\}$ satisfies conditions 1) and 2) for $A \otimes B$, then $\{(c_i, UC_iU^\dagger)\}$ also satisfies conditions 1) and 2) for $UAU^\dagger \otimes B$. This means we only need to prove this proposition with $A = |0\rangle\langle 0|^{n-m} \otimes I_m$ where $m \geq 0$.

First, note that

$$\begin{aligned} \sum_i c_i \langle 0|^{\otimes n-m} C_i |0\rangle^{\otimes n-m} &= \langle 0|^{\otimes n-m} A \otimes B |0\rangle^{\otimes n-m} \\ &= I_m \otimes B \end{aligned} \tag{45}$$

By Lemma 3, $c_i \langle 0|^{\otimes n-m} C_i |0\rangle^{\otimes n-m} = c_i' C_i'$ where $|c_i'| \leq |c_i|$ and C_i' are also stabilizer projectors. This means

$$\xi(I_m \otimes B) \leq \sum_i |c_i'| \leq \sum_i |c_i| = \xi(A \otimes B) \tag{46}$$

However, by the tensor invariance of ξ, *i.e.*,

$$\xi(A \otimes B) = \xi(|0\rangle\langle 0|^{\otimes n-m} \otimes I_m \otimes B) = \xi(I_m \otimes B),$$

we can conclude that $c_i = c_i'$. This means each C_i satisfies $C_i \sqsupseteq |0\rangle\langle 0|^{\otimes n-m} \otimes C_i'$ (see Lemma 3). Note that $\{(c_i, |0\rangle\langle 0|^{\otimes n-m} \otimes C_i')\}$ is also a v-optimal SPD for $A \otimes B$ since

$$\sum_i c_i |0\rangle\langle 0|^{\otimes n-m} \otimes C_i' = |0\rangle\langle 0|^{\otimes n-m} \otimes I_m \otimes B \tag{47}$$

and has smaller v^*, *i.e.*,

$$\sum_i |c_i| \mathrm{tr}(|0\rangle\langle 0|^{\otimes n-m} \otimes C_i') \leq \sum_i |c_i| \mathrm{tr}(C_i) \tag{48}$$

So by condition 2), we conclude that $C_i = |0\rangle\langle 0|^{\otimes n-m} \otimes C_i'$.

Next,

$$\begin{aligned} 2^m \cdot B &= \mathrm{tr}_m(I_m \otimes B) = \mathrm{tr}_m(\sum_i c_i C_i') \\ &= \sum_i c_i \mathrm{tr}_m(C_i') \end{aligned} \tag{49}$$

where tr_m denotes the partial trace for the system of first m qubits. By Lemma 4, $c_i \mathrm{tr}_m(C_i') = 2^m c_i'' C_i''$, where $|c_i''| \leq |c_i|$ and C_i'' are also stabilizer projectors. Thus $B = \sum_i c_i'' C_i''$ and

$$\xi(B) \leq \sum_i |c_i''| \leq \sum_i |c_i| = \xi(A \otimes B).$$

Similarly, by the tensor invariance of ξ, we can conclude that $c_i'' = c_i$. This means each C_i' has the form $C_i' = I_m \otimes C_i''$ (see Lemma 4). So,

$$C_i = |0\rangle\langle 0|^{\otimes n-m} \otimes I_m \otimes C_i'' = A \otimes C_i''.$$

□

Proposition 6. *Suppose A is a stabilizer projector and B is a positive operator. If $\mathcal{C} = \{(c_i, C_i)\} \in \mathrm{SPD}(A \otimes B)$ satisfies*

(1) *$\mathcal{C}$ is v^*-optimal,*
(2) *$\mathcal{C}$ has the smallest v among all v^*-optimal SPDs of $A \otimes B$*

then each C_i has the form $C_i = A \otimes C_i'$

Proof. Suppose A is an n-qubit stabilizer projector and let $m = \log \mathrm{tr} A$. Similarly, we only need to prove this proposition with $A = |0\rangle\langle 0|^{n-m} \otimes I_m$ where $m \geq 0$.

First, note that

$$\begin{aligned} \sum_i c_i \langle 0|^{\otimes n-m} C_i |0\rangle^{\otimes n-m} &= \langle 0|^{\otimes n-m} A \otimes B |0\rangle^{\otimes n-m} \\ &= I_m \otimes B \end{aligned} \tag{50}$$

By Lemma 3, $c_i \langle 0|^{\otimes n-m} C_i |0\rangle^{\otimes n-m} = c_i' C_i'$ where C_i' are stabilizer projectors. Note that

$$\begin{aligned} \mathrm{tr} C_i &= \sum_{j \in \{0,1\}^{n-m}} \mathrm{tr} \langle j | C_i | j \rangle \\ &\geq \mathrm{tr} \langle 0|^{\otimes n-m} C_i |0\rangle^{\otimes n-m} \\ &= \frac{c_i'}{c_i} \mathrm{tr} C_i' \end{aligned} \tag{51}$$

which means $|c_i|\mathrm{tr}C_i \geq |c_i'|\mathrm{tr}C_i'$. Then, we have

$$\xi^*(I_m \otimes B) \leq \sum_i |c_i'|\mathrm{tr}C_i' \leq \sum_i |c_i|\mathrm{tr}C_i = \xi^*(A \otimes B) \tag{52}$$

However, by the tensor invariance of ξ^*, *i.e.*,

$$\xi^*(A \otimes B) = \xi^*(|0\rangle\langle 0|^{\otimes n-m} \otimes I_m \otimes B) = \xi^*(I_m \otimes B).$$

We can conclude that $c_i\mathrm{tr}C_i = c_i'\mathrm{tr}C_i'$ and so that $\langle j|C_i|j\rangle = 0$ for all $0^{n-m} \neq j \in \{0,1\}^{n-m}$. This means $C_i = |0\rangle\langle 0|^{\otimes n-m} \otimes C_i'$.

Next, we have:

$$\begin{aligned} 2^m \cdot B = \mathrm{tr}_m(I_m \otimes B) &= \mathrm{tr}_m(\sum_i c_i C_i') \\ &= \sum_i c_i \mathrm{tr}_m(C_i') \end{aligned} \tag{53}$$

where tr_m denotes the partial trace for the system of first m qubits. By Lemma 4, $c_i\mathrm{tr}_m(C_i') = 2^m c_i'' C_i''$, where $|c_i''| \leq |c_i|$ and C_i'' are also stabilizer projectors. Thus $B = \sum_i c_i'' C_i''$ and $I_m \otimes B = \sum_i c_i'' I_m \otimes C_i''$. Note that

$$\sum_i |c_i''| 2^m \mathrm{tr}(C_i'') = \sum_i |c_i| \mathrm{tr}(C_i') = \xi(I_m \otimes B) \tag{54}$$

which means $\{(c_i'', I_m \otimes C_i'')\}$ is a ν^*-optimal SPD for $I_m \otimes B$. It also has smaller ν, *i.e.*,

$$\sum_i |c_i''| \leq \sum_i |c_i| \tag{55}$$

So by condition 2), we have $|c_i''| = |c_i|$, and thus $C_i' = I_m \otimes C_i''$ (see Lemma 4).

So, $C_i = |0\rangle\langle 0|^{\otimes n-m} \otimes I_m \otimes C_i'' = A \otimes C_i''$. □

Then, we are ready to prove Theorem 2. We will first prove the correctness (*i.e.*, the output of Algorithm 3 correctly reveal the locality) and then prove its optimality.

Proof of Theorem 2.

Correctness: First we prove the correctness of Algorithm 3. Note that V_1 maps $\hat{P}_1, \ldots, \hat{P}_l, P_U$ to $Z_1, \ldots, Z_l, Z_{l+1}$. This means the group $V_1 G_i V_1^\dagger$ can be generated by $Z_1, \ldots, Z_l, \tilde{P}_{i,1} \otimes P_{i,1}, \ldots, \tilde{P}_{i,l_i} \otimes P_{i,l_i}$, where $\tilde{P}_{i,j}$ are Pauli operators on first l qubits. Note that $\tilde{P}_{i,j}$ contains only I and Z, as $\tilde{P}_{i,j}$ commutes with all $Z_1, \ldots, Z_l$. We can set $\tilde{P}_{i,j}$ to I_l by multiplying several $Z_k, k \leq l$ if necessary. Thus each $V_1 G_i V_1^\dagger$ has the generators $Z_1, \ldots, Z_l, I_l \otimes P_{i,1}, \ldots, I_l \otimes P_{i,l_i}$, which means $V_1 A_i V_1^\dagger = |0\rangle\langle 0|^{\otimes l} \otimes A_i'$ where

$$A_i' = \prod_{j \leq l_i} \frac{I + P_{i,j}}{2}.$$

Besides, it holds that

$$V_1 U V_1^\dagger = V_1 e^{i\theta P_U} V_1^\dagger = e^{i\theta Z_{l+1}} = I_l \otimes e^{i\theta Z_1}.$$

Consider the subsystem that excludes the first l qubits. By steps 10-17 of Algorithm 3, we can ensure that each Q_i at most anti-commutes with one Q_j. Then, by exchanging the subscripts, we can make $\{Q_i\}$ normal, *i.e.*, $\exists t$ such that $Q_1, \ldots, Q_t$ commute with all Q_i for $i \leq s$ and Q_{t+2i-1} only anti-commutes with Q_{t+2i} for $i = 1, \ldots, (s-t)/2$. Then, we can find a Clifford unitary V_2 that maps $\{Q_1, \ldots, Q_t\}$ to $\{Z_1, \ldots, Z_t\}$ and maps $\{Q_{t+1}, \ldots, Q_s\}$ to $\{Z_{t+1}, X_{t+1}, \ldots, Z_{(t+s)/2}, X_{(t+s)/2}\}$ (see Lemma 1). Since $\{Q_1, \ldots, Q_s\}$ is the maximal independent set, V_2 transforms all

$P_{i,j}$ and Z_1 to the signed Pauli operators of the form $I_{n'-(s+t)/2} \otimes P$ where P acts on a subsystem of size $(s+t)/2$. Thus,

$$V_2 A_i' V_2^\dagger = I_{n'-(s+t)/2} \otimes A_i''$$

for all i and

$$V_2 U' V_2^\dagger = I_{n'-(s+t)/2} \otimes U'',$$

where A_i'' and U'' are the stabilizer projectors and non-Clifford unitary in a subsystem of size $(t+s)/2$. Note that $V = V_2 V_1$ is then the required unitary, *i.e.*,

$$VUV^\dagger = I_{n-(s+t)/2} \otimes U''$$
$$VA_iV^\dagger = |0\rangle\langle 0|^{\otimes l} \otimes I_{n'-(s+t)/2} \otimes A_i''.$$

Optimality: Then we prove the optimality of Algorithm 3. Suppose the optimal Clifford unitary is $\overline{V}$ which satisfies $\overline{V}A\overline{V}^\dagger = B \otimes \overline{A}$ and $\overline{V}U\overline{V}^\dagger = W \otimes \overline{U}$, where B is a stabilizer projector, W is a Clifford unitary and $\overline{A}, \overline{U}$ act in a subsystem of size $\overline{n}$. Without loss of generality, we can assume that $B = |0\rangle\langle 0|^{\otimes l} \otimes I_{n-\overline{n}-l}$ (by applying extra Clifford unitary on B). By conditions 1), 2) and Proposition 5 and 6, we know that for each A_i, it can be written as follows:

$$A_i = \overline{V}^\dagger (|0\rangle\langle 0|^{\otimes l} \otimes I \otimes \overline{A}_i)\overline{V} \tag{56}$$

where $\overline{A}_i$ are stabilizer projectors. For convenience, we denote $I_{n-\overline{n}-l} \otimes \overline{A}_i$ as $\tilde{A}_i$.

Besides, $W \otimes \overline{U}$ is of the form $I_{n-\overline{n}} \otimes e^{i\theta\overline{P}_U}$. This is because $e^{i\theta P_U} = U = \overline{V}^\dagger(W \otimes \overline{U})\overline{V}$ has two different eigenvalues (here we do not consider multiplicity of eigenvalues), which means W and $\overline{U}$ at most have two different eigenvalues. As $\overline{U}$ is a non-Clifford unitary, it has exactly two different eigenvalues λ_1, λ_2. If W has exactly two different eigenvalues μ_1, μ_2, then it can only be $\lambda_1 = -\lambda_2, \mu_1 = -\mu_2$. Thus $U = e^{i\theta P_U}$ has two eigenvalues $\pm\lambda_1\mu_1$, which means $\theta = \pm\pi/2$, contradicting that U is non-Clifford. So W has only one eigenvalue, which means $W = I$ (up to a global phase). Thus $\overline{U} = e^{i\theta\overline{P}_U}$ and $P_U = \overline{V}^\dagger(I \otimes \overline{P}_U)\overline{V}$. For convenience, we denote $I_{n-\overline{n}-l} \otimes \overline{P}_U$ as $\tilde{P}_U$.

Define $\overline{P}_i = \overline{V}^\dagger Z_i \overline{V}$ for $i = 1, \ldots, l$. Note that Z_i commutes with $\overline{P}_U$ for $i \le l$, so that $\overline{P}_i$ commutes with P_U for $i \le l$. And since all $Z_1, \ldots, Z_l$ are in the stabilizer group corresponding to $|0\rangle\langle 0|^{\otimes} l \otimes I \otimes \overline{A}_i$, by Equation (56), we have $\overline{P}_1, \ldots, \overline{P}_l \in G_1 \cap \ldots \cap G_m \equiv \hat{G}$ where G_i is the stabilizer group corresponding to A_i. Since $\overline{V}$ is optimal, $\langle \overline{P}_1, \ldots, \overline{P}_l \rangle$ is the maximal sub-group of $\hat{G}$ that commutes with P_U (otherwise there is some $P_{l+1} \in \hat{G}$ such that $\overline{P}_1, \ldots, \overline{P}_l, P_{l+1}, P_U$ are independent and commute with each other. Then we can find another Clifford unitary that maps $\overline{P}_1, \ldots, \overline{P}_l, P_{l+1}$ to $Z_1, \ldots, Z_{l+1}$ and $\overline{P}_U$ to Z_{l+2} to further shrink $\overline{A}$ and $\overline{U}$).

On the other hand, our algorithm exactly finds the unique maximal sub-group of $\hat{G}$ that commutes with P_U, *i.e.*, $\langle \hat{P}_1, \ldots, \hat{P}_l \rangle = \langle \overline{P}_1, \ldots, \overline{P}_l \rangle$. This is because if P_U commutes with $\hat{G}$, then $\hat{G}$ itself is the maximal sub-group. Otherwise suppose P_U does not commute with $\hat{G}$. If there is another maximal subgroup G' that commutes with P_U, let $P' \in G' - \langle \overline{P}_1, \ldots, \overline{P}_l \rangle$, then

$$|\langle \hat{P}_1, \ldots, \hat{P}_l, P' \rangle| = 2^{l+1} = |\hat{G}| \tag{57}$$

which means $\langle \hat{P}_1, \ldots, \hat{P}_l, P' \rangle = \hat{G}$ and thus $\hat{G}$ commutes with P_U, contradiction. Thus the V_1 found by our algorithm satisfies $V_1 A_i V_1^\dagger = |0\rangle\langle 0|^{\otimes l} \otimes A_i'$ and $V_1 P_U V_1^\dagger = Z_{l+1} = I_l \otimes Z_1$.

Note that V_1 and $\overline{V}$ are isomorphisms between $Z_1, \ldots, Z_l$ and $\langle \hat{P}_1, \ldots, \hat{P}_l \rangle = \langle \overline{P}_1, \ldots, \overline{P}_l \rangle$. Thus $V_1 \overline{V}^\dagger$ is an automorphism on $\langle Z_1, \ldots, Z_l \rangle$. Thus

$$\begin{aligned} |0\rangle\langle 0|^{\otimes l} \otimes A_i' &= V_1 A_i V_1^\dagger \\ &= V_1 \overline{V}^\dagger (|0\rangle\langle 0|^{\otimes l} \otimes \tilde{A}_i)(V_1 \overline{V}^\dagger)^\dagger \\ &= (|0\rangle\langle 0|^{\otimes l} \otimes I)\left[V_1 \overline{V}^\dagger (I \otimes \tilde{A}_i)(V_1 \overline{V}^\dagger)^\dagger\right] \end{aligned} \tag{58}$$

So

$$A_i' = \langle 0|^{\otimes l} V_1 \overline{V}^\dagger (I \otimes \tilde{A}_i)(V_1 \overline{V}^\dagger)^\dagger |0\rangle^{\otimes l} \tag{59}$$

for all i and

$$\begin{aligned} I_l \otimes Z_1 &= V_1 P_U V_1^\dagger \\ &= V_1 \overline{V}^\dagger (I \otimes \tilde{P}_U)(V_1 \overline{V}^\dagger)^\dagger \end{aligned} \tag{60}$$

From now on, we will use $\underline{P}$ to denote the unsigned Pauli operator by simply discarding the phase of P. If the underline is applied on a set $\underline{\{P, Q, \ldots\}}$, then it will denote $\{\underline{P}, \underline{Q}, \ldots\}$.

Consider these two sets of Pauli operators:

$$\begin{aligned} \overline{S} &= \text{span}(\underline{\text{SG}(\tilde{A}_1)}, \ldots, \underline{\text{SG}(\tilde{A}_m)}, \underline{\tilde{P}_U}) \\ S' &= \text{span}(\underline{\text{SG}(A_1')}, \ldots, \underline{\text{SG}(A_m')}, Z_1) \end{aligned} \tag{61}$$

where $\text{SG}(A)$ denotes the stabilizer group of the stabilizer projector A, "span" acts on the vector representations of the input and returns the set of (unsigned) Pauli operators corresponding to the spanned vector space. Then, we will show that $\overline{S}$ and S' are isomorphic.

Define the mapping $F : \overline{S} \to S'$ as

$$F(P) = \underline{\langle 0|^{\otimes l} V_1 \overline{V}^\dagger (I \otimes P)(V_1 \overline{V}^\dagger)^\dagger |0\rangle^{\otimes l}} \tag{62}$$

F has the following properties:

(1) F is bijective
(2) $P, Q \in \overline{S}$ commutes $\Leftrightarrow$ $F(P), F(Q)$ commutes.
(3) $\{P_1, \ldots, P_j\} \subseteq \overline{S}$ independent $\Leftrightarrow$ $\{F(P_1), \ldots, F(P_j)\}$ independent.

1) F is bijective:

Note that

$$\begin{aligned} &\underline{\overline{V}^\dagger \text{span}(Z_1, \ldots, Z_l, I_l \otimes \overline{S}) \overline{V}} \\ &= \text{span}(\underline{\text{SG}(A_1)}, \ldots, \underline{\text{SG}(A_m)}, \underline{P_U}) \\ &= \underline{V_1^\dagger \text{span}(Z_1, \ldots, Z_l, I_l \otimes S') V_1} \end{aligned} \tag{63}$$

This means $\text{span}(Z_1, \ldots, Z_l, I_l \otimes \overline{S})$ and $\text{span}(Z_1, \ldots, Z_l, I_l \otimes S')$ have the same size. And since

$$\langle Z_1, \ldots, Z_l \rangle \cap (I_l \otimes \overline{S}) = \langle Z_1, \ldots, Z_l \rangle \cap (I_l \otimes S') = \emptyset,$$

$I_l \otimes \overline{S}$ and $I_l \otimes S'$ have the same size, so do $\overline{S}$ and S'. Equation (59) implies that F is a surjection of $\underline{\mathrm{SG}(\tilde{A}_i)} \to \underline{\mathrm{SG}(A'_i)}$ for all i. Thus F is surjection of $\overline{S} \to S'$. And since $|\overline{S}| = |S'|$, we conclude that F is a bijection of $\overline{S} \to S'$.

2) $P, Q \in \overline{S}$ commutes $\Leftrightarrow$ $F(P), F(Q)$ commutes:

Equation (59) implies that for all i and $P \in \mathrm{SG}(A_i)$, $\underline{V_1\overline{V}^\dagger(I\otimes P)(V_1\overline{V}^\dagger)^\dagger}$ is of the form $\perp_1 \otimes \cdots \otimes \perp_l \otimes P'$, where $\perp_j \in \{I, Z\}$ (since otherwise the RHS of Equation (59) equals cA where $c < 1$ and A is a stabilizer projector, see also the proof of Lemma 2 and Lemma 3). Combined with Equation (60), we conclude that for all $P \in \overline{S}$, $\underline{V_1\overline{V}^\dagger(I\otimes P)(V_1\overline{V}^\dagger)^\dagger}$ is of the form $\perp_1 \otimes \cdots \otimes \perp_l \otimes P'$, where $\perp_i \in \{I, Z\}$. We have

$$\forall P \in \overline{S}, \quad \underline{V_1\overline{V}^\dagger(I\otimes P)(V_1\overline{V}^\dagger)} = \perp_P \otimes F(P) \tag{64}$$

So

$$\begin{aligned} & P, Q \text{ commutes} \\ \Leftrightarrow\ & \underline{V_1\overline{V}^\dagger(I\otimes P)(V_1\overline{V}^\dagger)^\dagger}, \underline{V_1\overline{V}^\dagger(I\otimes Q)(V_1\overline{V}^\dagger)^\dagger} \text{ commutes} \\ \Leftrightarrow\ & \perp_P \otimes F(P), \perp_Q \otimes F(Q) \text{ commutes} \end{aligned} \tag{65}$$

Since $\perp_P, \perp_Q$ always commutes,

$$P, Q \text{ commutes} \ \Leftrightarrow\ F(P), F(Q) \text{ commutes} \tag{66}$$

3) $\{P_1, \ldots, P_j\} \subseteq \overline{S}$ independent $\Leftrightarrow$ $\{F(P_1), \ldots, F(P_j)\}$ independent:

For $\Leftarrow$ part, we have:

$$\begin{aligned} & \{F(P_i)\} \text{ is independent} \\ \Rightarrow\ & \{\perp_{P_i} \otimes F(P_i)\} = \{\underline{V_1\overline{V}^\dagger(I\otimes P_i)(V_1\overline{V}^\dagger)^\dagger}\} \text{ is independent} \\ \Rightarrow\ & \{P_i\} \text{ is independent} \end{aligned} \tag{67}$$

For the $\Rightarrow$ part, we have:

$$\begin{aligned} \underline{(\perp_P \otimes F(P))(\perp_Q \otimes F(Q))} &= \underline{V_1\overline{V}^\dagger PQ(V_1\overline{V}^\dagger)^\dagger} \\ &= \underline{\perp_P \perp_Q} \otimes F(\underline{PQ}) \end{aligned} \tag{68}$$

Thus

$$\underline{F(P)F(Q)} = F(\underline{PQ}) \tag{69}$$

Suppose $\{P_i\}$ is independent but $\{F(P_i)\}$ is not. Without loss of generality, we can assume $\underline{F(P_1)\cdots F(P_j)} = I$. Then $F(\underline{P_1 \cdots P_j}) = I$. However, $F(I) = I$ and $\underline{P_1 \cdots P_j} \neq I$. This contradicts the fact that F is bijective.

At this point, we can conclude that F satisfies the properties 1), 2) and 3). Thus F is an isomorphism between $\overline{S}$ and S'.

Suppose $\bar{s}$ is the dimension of $\overline{S}$. By Lemma 5, there is a maximal independent $\bar{t}$-normal set (see the Definition 4) $\{P_1, \ldots, P_{\bar{s}}\} \subseteq \overline{S}$ for some $\bar{t}$. Note that each $P \in \overline{S}$ is of the form $I_{n-\overline{n}-l} \otimes \perp_P$, where $\perp_P$ is an $\overline{n}$-qubit Pauli operator. So there are $(\bar{s}+\bar{t})/2$ independent $\overline{n}$-qubit Pauli operators $\perp_{P_1}, \ldots, \perp_{P_{\bar{t}-1}}, \perp_{P_{\bar{t}}}, \perp_{P_{\bar{t}+2}}, \perp_{P_{\bar{t}+4}}, \ldots, \perp_{P_{\bar{s}}}$ that commute with each other. This implies that $\overline{n} \geq (\bar{s}+\bar{t})/2$.

By properties 2) and 3) of F, $\{F(P_1), \ldots, F(P_{\overline{s}})\}$ is also a maximal independent $\overline{t}$-normal set of S'. Our algorithm finds another maximal independent t-normal set $\{h_1, \ldots, h_s\}$ of S'. Since both of them are maximal independent sets of S', we have $s = \overline{s}$. By Lemma 6, we have $t = \overline{t}$. So our algorithm find the unitary V_2 that maps each A'_i and P'_U into a subsystem of $(s+t)/2 = (\overline{s}+\overline{t})/2 \leq \overline{n}$. This proves the optimality of our algorithm. □

Definition 4. *We say that a set of Pauli operators $\{P_1, \ldots, P_s\}$ is t-normal if*

(1) *$P_1, \ldots, P_t$ commute with all $P_i, 1 \leq i \leq s$*
(2) *P_{t+2i-1} only anti-commutes with P_{t+2i} for $1 \leq i \leq (s-t)/2$*

Lemma 5. *Suppose S is a set of Pauli operators that satisfies $\forall P, Q \in S$, $\underline{PQ} \in S$. Then, there is a independent subset $\{P_1, \ldots, P_s\} \subseteq S$ such that*

(1) *$S = \text{span}(P_1, \ldots, P_s)$,*
(2) *$\{P_1, \ldots, P_s\}$ is t-normal for some t*

We say that $\{P_1, \ldots, P_s\}$ is a maximal independent t-normal set.

Proof. First find a maximal independent subset $P_1, \ldots, P_s \subseteq S$, then use the similar manner as steps 10-17 of Algorithm 3 and exchange the subscripts if necessary. □

Lemma 6. *Suppose S is a set of Pauli operators that satisfies $\forall P, Q \in S$, $\underline{PQ} \in S$. $\{P_1, \ldots, P_s\}$ is a maximal independent t_1-normal set of S and $\{Q_1, \ldots, Q_s\}$ is a maximal independent t_2-normal set of S. Then $t_1 = t_2$.*

Proof. Suppose $t_1 < t_2$. Since both $\{P_i\}$ are maximal independent set, each Q_i can be represented as a combination of $P_1, \ldots, P_s$. Note that there must be a $Q \in \{Q_1, \ldots, Q_{t_2}\}$ that has components from $P_{t_1+1}, \ldots, P_s$ (otherwise $\{Q_1, \ldots, Q_{t_2}\}$ can be represented by only $\{P_1, \ldots, P_{t_1}\}$ and thus cannot be independent). Let this Q be Q_1 and

$$Q_1 = \underline{Q'_1 \prod_{i \in X} P_i}$$

where $\emptyset \neq X \subseteq \{t_1 + 1, \ldots, s\}$ and Q'_1 is composed of elements from $\{P_1, \ldots, P_{t_1}\}$. Define the set Y to be

$$\begin{aligned} Y = \{i | \exists j \in X, k \geq 1,\ & (i,j) = (t_1 + 2k - 1, t_1 + 2k) \\ & \vee\ (j,i) = (t_1 + 2k - 1, t_1 + 2k)\} \end{aligned} \tag{70}$$

Since X is non-empty, Y is also non-empty. Then, suppose each $Q_i, i \geq 2$ has the form

$$Q_i = \underline{Q'_i \prod_{j \in Y_i} P_j}$$

where $Y_i \subseteq Y$ and Q'_i is composed of elements from $\{P_k | k \notin Y\}$.

First note that $|Y_i|$ must be even. Note that all $Q_i, i \geq 2$ commute with Q_1, which means $\underline{\prod_{j \in Y_i} P_j}$ commutes with $\underline{\prod_{j \in X} P_j}$. And whether $\underline{\prod_{j \in Y_i} P_j}$ commutes with $\underline{\prod_{j \in X} P_j}$ only depends on the parity of how many pairs of $(P_j, P_k), j \in X, k \in Y_i$ that are anti-commute. On the other hand, by the definition of Y, we have

$$\forall k \in Y, \exists! j \in X, \quad P_j \text{ anti-commutes with } P_k \tag{71}$$

This implies that $|Y_i|$ must be even to ensure that there are even pairs of $(P_j, P_k), j \in X, k \in Y_i$ that are anti-commute.

Besides, $|X \cap Y|$ is always even. This is because, if $z = t_1 + 2k \in X \cap Y$ then $z - 1 = t_1 + 2k - 1 \in X \cap Y$, and if $z = t_1 + 2k - 1 \in X \cap Y$ then $z + 1 = t_1 + 2k \in X \cap Y$. Thus all elements in $X \cap Y$ can be paired to $(2k-1, 2k)$ for some $k \geq 1$.

Thus we conclude that for all i, Q_i has exactly even components from $P_Y = \{P_j | j \in Y\}$. Note that, $\{Q_i\}$ is maximal independent, which means for any $P \in P_Y$, P is representable by a combination of $\{Q_i\}$. But any Q_i has even components from P_Y, so that the combination of $\{Q_i\}$ cannot represent P. Contradiction!

Thus $t_1 \geq t_2$, and similarly we also have $t_2 \geq t_1$, which means $t_1 = t_2$. □