Synthesis of fault-tolerant state preparation circuits using Steane-type error detection

Erik Weilandt [1,*] Tom Peham [1] and Robert Wille [1,2]
[1]*Chair for Design Automation, Technical University of Munich, Munich, Germany*
[2]*MQSC, Garching near Munich, Germany*

(Received 20 February 2026; revised 28 August 2026; accepted 2 October 2026; published 13 November 2026)

Fault-tolerant state preparation is essential for reliable quantum error correction, particularly in Steane-type error correction, which relies on robust ancilla states for syndrome readout. One method of fault-tolerant state preparation is to initialize multiple ancilla states and check them against each other to detect problematic errors. In the worst case, the number of states required for successful initialization grows polynomially with the code distance, but it has been shown that this can be reduced to constant ancilla overhead; in the best case, only four states are required. However, existing techniques for finding low-overhead initialization schemes are limited to codes with large symmetry groups, such as the GOLAY code. In this work we propose a general, automated synthesis methodology for Steane-type fault-tolerant state preparation circuits that applies to arbitrary Calderbank-Shor-Steane (CSS) codes and does not rely on code symmetries. We apply the proposed methods to various CSS codes up to a distance of 7 and simulate the successful fault-tolerant initialization of logical basis states under circuit-level depolarizing noise. The circuits synthesized using the proposed methodology provide an important step towards experimental realizations of high-fidelity ancilla states for near-term demonstration of fault-tolerant quantum computation.

DOI: 10.1103/y9ln-b4t9

I. INTRODUCTION

A major challenge on the path towards utility-scale quantum computing is the noise inherent in any quantum system. Fault-tolerant quantum computing (FTQC) and quantum error correction (QEC) address this issue by encoding quantum information redundantly in a quantum error-correction code [1–3]. Calderbank-Shor-Steane (CSS) codes [4–6] are a prominent class of stabilizer codes [7] that are constructed from two classical linear codes which allow for independently detecting and correcting bit flip (Pauli X errors) and phase flip (Pauli Z errors).

An essential property of CSS codes is that any CSS code admits a transversal controlled-NOT (CNOT) gate, which means that a logical CNOT gate between the qubits of two code blocks can be implemented via physical CNOT gates between the physical qubits of the respective blocks. Transversal gates are desirable for FTQC because errors in those gates do not spread between data qubits of the same block. In Steane-type error correction [8,9], transversal CNOT gates are used to copy errors from an encoded data block to an ancillary system that encodes the logical all-zero $|0\rangle_L^{\otimes k}$ or logical all-plus $|+\rangle_L^{\otimes k}$ state. Measurements of the ancilla system are then used to infer and correct errors on the data qubits. Since the ancilla system can be initialized independently and has only minimal interaction with the data block, Steane-type QEC is a promising candidate for practical quantum error correction, especially for quantum computing architectures that exhibit a high degree of gate-level parallelism like trapped-ion quantum computers [10–12] or neutral-atom quantum computers [13–16].

The ancillary systems used for Steane-type QEC should have a high fidelity to avoid introducing further errors on the data. This necessitates robust, fault-tolerant state preparation circuits. One way to create high-fidelity states is via a non-deterministic repeat-until-success protocol, where the state is initialized with a non-fault-tolerant unitary encoding circuit augmented by an error-detection gadget. The initialization is then repeated until the error-detection circuit measurements indicate that no problematic errors occurred during state preparation.

One way to implement the error-detection circuit is to measure specific stabilizers of the code [16–19]. While manageable for small codes, the number of measurements required to detect all problematic errors quickly becomes large. Hence, it is increasingly more difficult to find such a minimal set of measurements [18]. Furthermore, high-weight stabilizer measurements need to be protected by flag qubits [20,21] to avoid introducing hook errors on the data qubits. Yet another method is to incorporate flag qubits directly into the unitary encoding circuit [22]. This approach reduces circuit overhead for error detection, since errors are detected within specific subcircuits during encoding rather than at the end of the circuit. However, this method requires a specific circuit structure with significant depth overhead.

An alternative state preparation approach is illustrated in Fig. 1(a). One prepares multiple ancilla states, copies potential errors from one ancilla state to another using a transversal CNOT gate, and detects them by measuring the second state.

*Contact author: erik.weilandt@tum.de

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

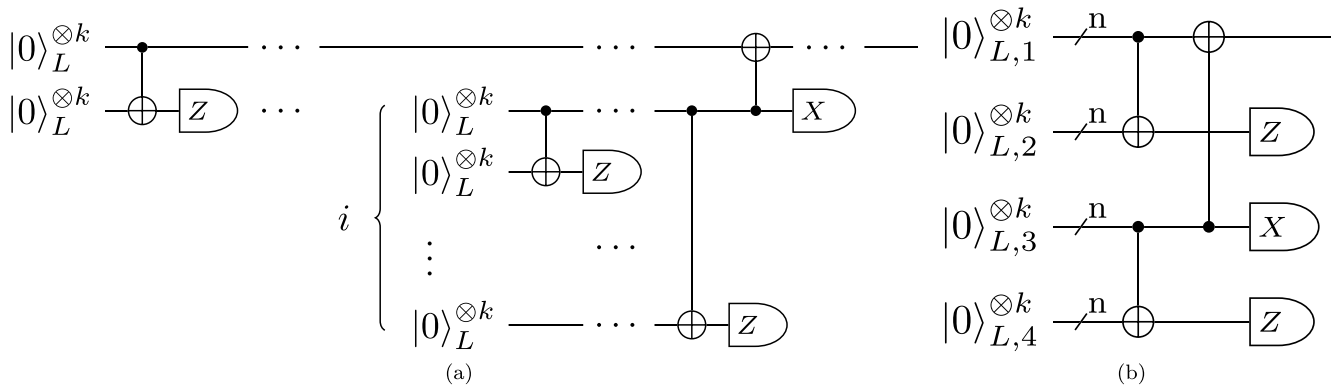


FIG. 1. Nondeterministic Steane-type state preparation. (a) The state is prepared by non-fault-tolerantly preparing multiple states and copying errors on data qubits over to ancilla qubits, which are measured out in the respective basis for error detection. This process is repeated until no errors are detected on any of the states. This process requires $O(nd^2)$ physical qubits and measurements if the states are prepared using the same circuit, since the ancilla needs to be recursively verified using additional ancillas. (b) Four states are prepared using different state preparation circuits. This construction is fault tolerant if the sets of propagated errors of all circuits are sufficiently different.

As with Steane-type QEC, this has the benefit of a low-depth overhead since the ancilla can be prepared in parallel. The issue is that specific errors might cancel out when they are copied with the transversal CNOT gate, effectively lowering the distance of the code. Consequently, multiple verifications are required to ensure no such errors are undetected.

Reference [23] shows how this ancilla overhead can be made constant for the GOLAY code by preparing four states using different preparation circuits that guarantee that high-weight errors do not go by undetected due to cancellations on the logical ancilla qubits as illustrated in Fig. 1(b). This approach is based on qubit permutations using qubit symmetries but has been shown only for the highly symmetrical GOLAY code [23], REED-MULLER codes [24], and Greenberger-Horne-Zeilinger ancilla states [25] thus far; no general circuit-synthesis methodology has been proposed for constructing constant-overhead Steane-type fault-tolerant state preparation (FTSP) circuits.

This work closes this gap by proposing a general circuit synthesis methodology for this state preparation scheme. We show that constructing the desired circuits does not rely on code symmetries. In fact, the problem can be addressed by using different CNOT circuits that prepare the same state but propagate errors differently. Code symmetries are then only one specific way of obtaining such circuits.

Our main contributions are as follows.

(i) We explicitly give manual constructions for fault-tolerant Steane-type state preparation circuits for the $[\![19, 1, 5]\!]$ code, which has only a few qubit symmetries, and the $[\![17, 1, 5]\!]$ code, which has no qubit symmetries at all.

(ii) For circuits with combinatorially more challenging error sets, we utilize a heuristic CNOT circuit synthesis that constructs multiple circuits with different fault sets on the fly.

(iii) Using the proposed automated synthesis method, we construct several low-depth Steane-type fault-tolerant state preparation schemes for codes up to distance 7.

The proposed methodology is intended for the regime of small- and medium-distance codes where explicit enumeration of fault sets remains tractable. This regime aligns with current and near-term experimental demonstrations of fault-tolerant quantum computing, where codes of distances up to 7 are being explored on trapped-ion and neutral-atom platforms. For larger distances, the size of the fault sets to be checked during synthesis grows quickly and the method becomes increasingly difficult to apply.

We compare the resulting circuit designs with state preparation circuits from Ref. [18] as well as Ref. [22] and provide numerical evidence for the fault-tolerance properties of the synthesized circuits via Monte Carlo simulation. All proposed methods and circuits are publicly available in the form of open-source software as part of the Munich Quantum Toolkit (MQT) [26].

The rest of this paper is structured as follows. Section II provides the necessary background on FTQC and introduces the noise model and fault-tolerance criteria considered in this work. Section III reviews and motivates the problem of circuit synthesis for Steane-type fault-tolerant state preparation. Section IV shows explicit constructions for distance-5 color codes. Section V proposes an automated synthesis method for constructing Steane-type fault-tolerant state preparation circuits. Section VI shows the results of conducted simulations, which show that the proposed constant low-overhead constructions are strictly fault tolerant. Section VII provides a summary and discusses prospects for future work.

II. BACKGROUND

This section provides background on the relevant concepts of quantum error correction and establishes the notation used in this work.

Throughout this work we use $[i] = \{1, 2, \ldots, i\} \subseteq \mathbb{Z}$.

A. CSS codes

An $[\![n, k, d]\!]$ stabilizer code on n qubits is defined by a stabilizer group $\mathcal{S}$, a commutative subgroup of the n-qubit Pauli group $\mathcal{P}_n$, which encodes k logical qubits. The code space of a stabilizer code is defined as the simultaneous $+1$ eigenspace of all elements of the stabilizer group. The distance of the code is defined as the minimal-weight Pauli operator that acts

nontrivially on the code space, where we define the weight $\mathrm{wt}(P)$ of a Pauli operator P on n qubits as the number of qubits on which it acts nontrivially. We say that two Pauli operators P_1 and P_2 are stabilizer equivalent (or equivalent up to stabilizers) with respect to $\mathcal{S}$, denoted by $P_1 \simeq_{\mathcal{S}} P_2$, if there is an element $S \in \mathcal{S}$ such that $SP_1 = P_2$.

A stabilizer code is CSS if it is the combination of two groups $\mathcal{S}_X$ and $\mathcal{S}_Z$, where all elements of $\mathcal{S}_X$ ($\mathcal{S}_Z$) are tensor products of Pauli X (Z) and the identity I. The groups $\mathcal{S}_X$ and $\mathcal{S}_Z$ can be identified with $\mathbb{F}_2$ vector spaces and are often described via the check matrices H_X and H_Z whose rows are a basis of the respective vector spaces. The CSS condition states that $H_X H_Z^T = 0$, which expresses the notion that all elements of $\mathcal{S}_X$ and $\mathcal{S}_Z$ commute.

A detectable error (or just an error) is a Pauli operator that moves a code word out of the code space. For CSS codes, we usually treat X-type and Z-type errors separately, even if the noise includes combinations of both. This is because CSS codes can detect and correct X- and Z-type errors independently, and any mixed error can be broken down into a combination of these two types and the identity I. An X error $E \in \mathbb{F}_2^n$ is detected by the Z checks, i.e., $H_Z E \neq 0$ (similar for Z errors and X checks). Logical X and Z operators L_X and L_Z are defined as operators that commute with all stabilizers and anticommute with each other.

B. Circuit-level noise model

This work considers quantum circuits composed of physical qubit initializations in $|0\rangle$ and $|+\rangle$, CNOT gates between arbitrary qubits, and measurements in the X or Z basis. We employ a circuit-level noise model parametrized by a physical error rate p. Noisy gates are modeled as ideal gates followed by a depolarizing noise channel parametrized by a physical noise rate p:

$$\epsilon_2(\rho) = (1-p)\rho + \frac{p}{15}\sum_{E\in\mathcal{E}} E\rho E.$$

Here $\mathcal{E} = \{P_1 \otimes P_2 \mid P_1, P_2 \in \{I, X, Y, Z\}\} \setminus \{I \otimes I\}$. In addition, qubits are initialized in the -1 eigenstate of the respective basis with probability $O(p)$, and measurements also flip with a probability of $O(p)$. If a qubit is idle during one layer of CNOT gates, it is also subject to single-qubit depolarizing idling noise of strength $O(p)$.

The notion of the types of errors that can propagate through a circuit via two-qubit gates is central to the following discussion.

Definition 1. Let $\mathcal{C}$ be a quantum circuit with n qubits and m gates and let $\mathcal{C}_j$ denote the circuit obtained from $\mathcal{C}$ by removing the first j gates. The X (Z) fault set $\mathcal{E}_X(\mathcal{C})$ [$\mathcal{E}_Z(\mathcal{C})$] of $\mathcal{C}$ is the set of errors stemming from a single propagated X (Z) fault in $\mathcal{C}$. More precisely, $\mathcal{E}_X(\mathcal{C}) = \{\mathcal{C}_j X_i \mathcal{C}_j^{-1} \mid i \in [n], j \in [m]\}$.

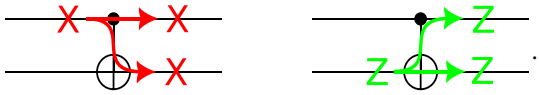
The minimal weight of an error E is defined as $\min_{S\in\mathcal{S}} \mathrm{wt}(SE)$. We will generally omit the stabilizer group $\mathcal{S}$ and, since we are only concerned with minimal representatives, refer to the minimal weight simply as $\mathrm{wt}(E)$.

As we aim to eventually construct fault-tolerant state preparation circuits, we define strict fault tolerance: A circuit should not amplify errors during execution.

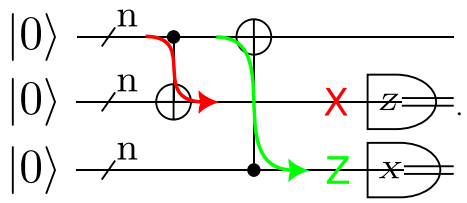
Definition 2. We say that a state preparation circuit $\mathcal{C}$ for an $[\![n, k, d]\!]$ code is strictly fault tolerant if there are no errors $e_1, \ldots, e_t \in \mathcal{E}_X(\mathcal{C})$ [$\mathcal{E}_Z(\mathcal{C})$] for $t \leqslant \lfloor (d-1)/2 \rfloor$ such that $\mathrm{wt}(\sum_{i=1}^{t} e_i) > t$.

III. PROBLEM DEFINITION AND MOTIVATION

We consider the problem of initializing the logical $|0\rangle_L^{\otimes k}$ state of an $[\![n, k, d]\!]$ CSS code. One way to achieve this is by initializing the n physical qubits to $|0\rangle$ and $|+\rangle$ and executing a unitary CNOT encoding circuit to entangle these qubits into the desired state. Such an encoding circuit is generally not fault tolerant, as single Pauli errors in the circuit can propagate through a CNOT gate:



One approach to making unitary state preparation resilient to such errors is by initializing two copies of a state, using one state to verify whether errors have occurred in the other state. In the case of CSS codes, this can be done with a simple circuit using transversal CNOT gates and measurement of two ancilla blocks, one measured in the Z basis to detect X-type errors and one in the X basis to detect Z-type errors:



Since a transversal CNOT gate on a CSS code implements a logical CNOT gate between all encoded qubits of two code blocks, this circuit acts trivially on the code space. However, errors still propagate through the physical CNOT gates. These errors can be detected in the ancilla measurements. Therefore, this circuit can be used in a repeat-until-success scheme by postselecting on these measurements. We refer to such a state preparation scheme as Steane-type (nondeterministic) fault-tolerant state preparation.

Example 1. Consider Steane-type preparation of the logical state $|0\rangle_L$ for the Steane code, as shown in Fig. 2. In this scenario, the data block (block I) undergoes a single-qubit error X_1, which propagates within block I to a weight-2 error. During the subsequent transversal CNOT operation (block III), this error is further transferred from the data block to the ancilla block (block II). As a result, the error remains on data qubits q_1 and q_3 and on the corresponding ancilla qubits q_8 and q_{10}, where it is ultimately detected through measurement of the ancilla.

Depending on how the state preparation circuits are chosen in Steane-type FTSP, the protocol may fail to be fault tolerant. The issue arises when the same preparation circuit is used for both the data and the ancilla blocks: Identical faults occurring at the same circuit location can propagate through the transversal CNOT in such a way that they cancel on the ancilla. In this situation, the ancilla measurements reveal no syndrome, even though a nontrivial error remains on the data

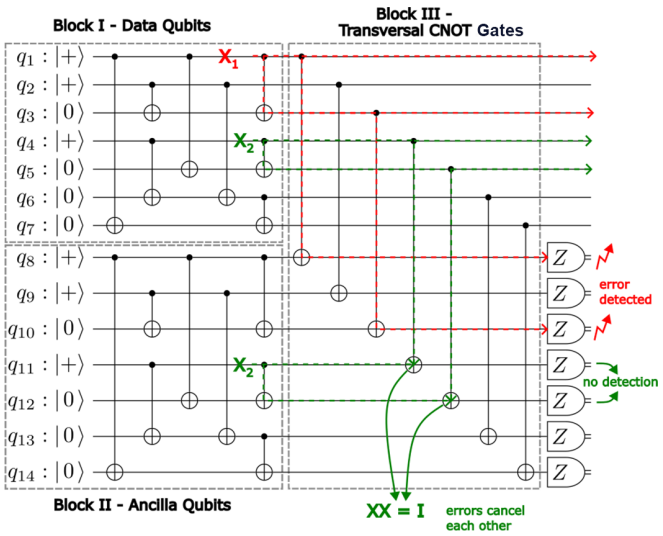


FIG. 2. Blocks I and II are state preparation circuits for the logical zero state of the code. Block I represents the data structure, and block II is the ancilla structure. Block III is the connection of both state preparation circuits through transversal CNOT gates. Together, all three blocks form the base structure of Steane-type error correction.

block. Thus, the data error goes undetected, breaking fault tolerance.

Example 2. Consider the circuit in Fig. 2 again. If an X_2 error occurs at the same location in both state preparation circuits, it propagates through the transversal CNOT gate from the data block to the ancilla block, where it cancels with the identical error already present on the ancilla. As a result, the ancilla measurements reveal no syndrome, leaving the error on the data block undetected.

While this cancellation does not affect the Steane code, which only protects against single-qubit errors, it becomes problematic for larger, higher-distance codes, where such undetected cancellations violate strict fault tolerance.

A brute-force way to compensate for such errors is to use additional ancillas and repeat the transversal CNOT gate and measurement multiple times. Unfortunately, the ancillas used to verify the data can introduce high-weight errors, so they would need to be checked by additional ancillary qubits. In fact, ancillas need to be recursively verified until it is ensured that all occurrences of $t < \frac{d-1}{2}$ independent errors can be detected. This is the protocol depicted in Fig. 1(a).

With increasing distance, this protocol quickly results in circuits with high depth and CNOT gate count since many states are required for the recursive verification scheme. Because each additional ancilla block adds CNOT gates and qubits, each of which is itself a source of new faults, the recursive verification scheme has a large number of fault locations. Moreover, each level of recursion further reduces the acceptance rate. This extremely pessimistic scheme ensures that, even if errors cancel out on an ancilla, another check will still detect the error.

Reference [23] shows that four state preparation circuits are sufficient for Steane-type FTSP, provided the circuits are chosen so that their fault sets are sufficiently distinct to prevent the problematic error cancellations described above [see Fig. 1(b)]. The notion of sufficiently distinct fault sets is formalized below: No nontrivial error can cancel during any transversal CNOT gate. In this construction, the measurement of the second ancilla $|0\rangle_{L,2}^{\otimes k}$ detects all X-type faults that have propagated from the first state $|0\rangle_{L,1}^{\otimes k}$. Similarly, the third ancilla $|0\rangle_{L,3}^{\otimes k}$ detects all propagated Z-type faults. Finally, the fourth ancilla $|0\rangle_{L,4}^{\otimes k}$ detects any X faults on $|0\rangle_{L,3}^{\otimes k}$, ensuring that the Z-syndrome extraction on $|0\rangle_{L,1}^{\otimes k}$ does not introduce new undetected errors. If all three verification steps succeed, the remaining state $|0\rangle_{L,1}^{\otimes k}$ has residual error weight at most $(d-1)/2$, provided no more than $(d-1)/2$ faults occurred during preparation. This guarantee is achieved while requiring only a constant number of ancillas.

Achieving this constant overhead requires choosing circuits whose fault sets are sufficiently distinct. To formalize what it means for two circuits to be "different enough," we introduce the notion of t-distinctness. Intuitively, two fault sets are t-distinct if no combination of at most t faults from each set can produce equivalent errors on both circuits, preventing cancellations under transversal interaction, and if any such combination necessarily has weight larger than the number of faults involved.

Definition 3. Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two fault sets and $\mathcal{S}$ be a stabilizer group. Then we say that $\mathcal{E}_1$ and $\mathcal{E}_2$ are t-distinct with respect to $\mathcal{S}$ if there are no subsets $\mathcal{F}_1 \subseteq \mathcal{E}_1$ and $\mathcal{F}_2 \subseteq \mathcal{E}_2$ with $|\mathcal{F}_1| + |\mathcal{F}_2| \leqslant t$ such that $\prod_{e\in\mathcal{F}_1} e \simeq_{\mathcal{S}} \prod_{e\in\mathcal{F}_2} e$ and $\mathrm{wt} \prod_{e\in\mathcal{F}_1} e > |\mathcal{F}_1| + |\mathcal{F}_2|$.

At first glance, one might expect that all fault sets of the circuits $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ preparing $|0\rangle_{L,1}$, $|0\rangle_{L,2}$, $|0\rangle_{L,3}$, and $|0\rangle_{L,4}$ must be $\lfloor (d-1)/2 \rfloor$-distinct in order for the scheme in Fig. 1(b) to be strictly fault tolerant. However, this is not necessary. Examining how errors propagate and interact in Fig. 1(b) shows that only certain pairs of fault sets must satisfy $\lfloor (d-1)/2 \rfloor$-distinctness.

We say that four state preparation circuits $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, and $\mathcal{C}_4$ for an $[\![n, k, d]\!]$ CSS code implement the protocol in Fig. 1(b) in a strictly fault-tolerant manner if the following pairs of fault sets are $\lfloor (d-1)/2 \rfloor$-distinct:

$$\begin{aligned} &\mathcal{E}_X(\mathcal{C}_1), && \mathcal{E}_X(\mathcal{C}_2) && \text{(requirement 1)},\\ &\mathcal{E}_X(\mathcal{C}_3), && \mathcal{E}_X(\mathcal{C}_4) && \text{(requirement 2)},\\ &\mathcal{E}_Z(\mathcal{C}_1) \cup \mathcal{E}_Z(\mathcal{C}_2), && \mathcal{E}_Z(\mathcal{C}_3) \cup \mathcal{E}_Z(\mathcal{C}_4) && \text{(requirement 3)}. \end{aligned} \tag{1}$$

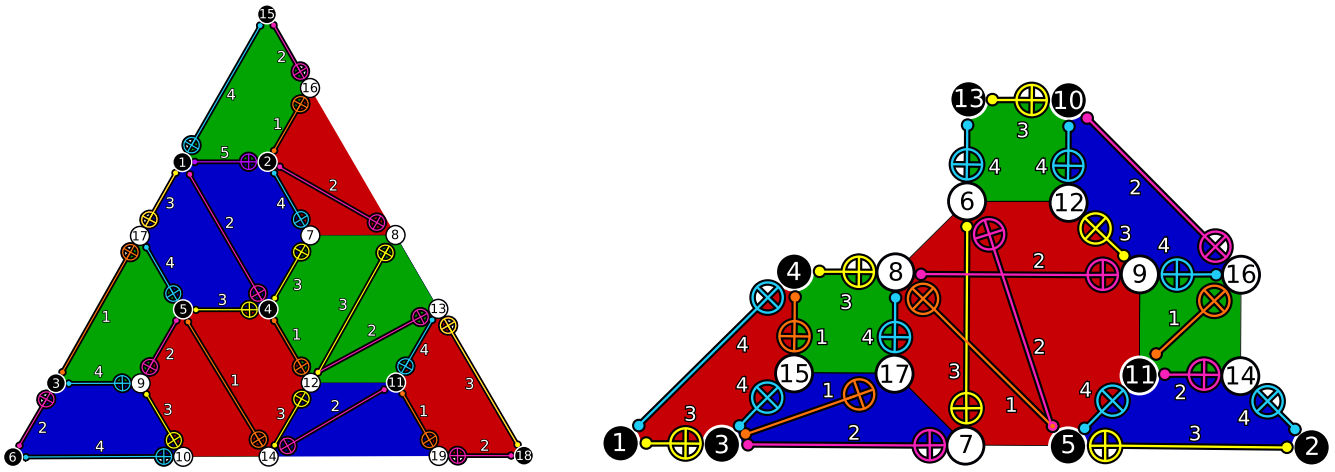
This work aims to construct such circuits for arbitrary CSS codes, as summarized by the following problem statement.

Problem 1. Circuit synthesis for Steane-type FTSP.

Input. Stabilizer generators of an $[\![n, k, d]\!]$ CSS code.

Problem. Construct four state preparation circuits $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, and $\mathcal{C}_4$ for $|0\rangle_L$ that implement the nondeterministic FTSP shown in Fig. 1(b).

The t-distinctness requirement necessitates leveraging the degrees of freedom in circuit construction to alter the circuits' fault sets. In Ref. [23], four 3-distinct state preparation circuits are constructed for the Golay code [27] by exploiting the Golay code's large symmetry group, the Mathieu group M_{23}. The circuits are obtained by taking a circuit $\mathcal{C}_1$ and deriving the three other circuits by applying certain symmetries of the code. Such permutations preserve the encoded logical



(a) (b)

FIG. 3. State preparation circuits for the $|0\rangle_L$ of the (a) $[\![19, 1, 5]\!]$ and (b) $[\![17, 1, 5]\!]$ color codes [28]. Black vertices represent qubits prepared in $|+\rangle$, and white vertices represent qubits prepared in $|0\rangle$. The CNOT gate order is indicated by the edge colors (orange, magenta, yellow, blue, and purple).

information while altering the specific fault set associated with a state preparation.

A key limitation of this approach is that it relies on large symmetry groups to achieve t-distinctness. While powerful, this technique does not apply to all CSS codes, as many relevant codes lack sufficiently large symmetry groups. Consequently, a more generalized approach is required, one that does not rely on code symmetries but achieves t-distinctness by appropriately constructing the CNOT circuits. In this work we will show how this can be achieved in an automated fashion for codes of moderate size that are suitable for Steane-type FTSP.

IV. MANUAL CONSTRUCTIONS FOR STEANE-TYPE FAULT-TOLERANT STATE PREPARATION

It is impractical to independently synthesize the circuits $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ in order to solve Problem 1, because the t-distinctness constraints tightly couple their fault sets. Intuitively, a much better approach is to take a reference circuit, say, $\mathcal{C}_1$, and generate the other circuits by altering this reference circuit.

The following two examples show how symmetries or local circuit rewrites can be used to obtain circuits with suitably distinct fault sets. From this we derive circuit rewriting techniques that allow us to synthesize Steane-type FTSP circuits for higher-distance codes that are hard to reason about manually.

A. $[\![19, 1, 5]\!]$ color code

The $[\![19, 1, 5]\!]$ color code shown in Fig. 3(a) is obtained from a hexagonal tiling of the plane. Qubits are placed on the vertices of a triangular region of the tiling, and the red, green, and blue plaquettes represent both X and Z stabilizers. This code's symmetry group is generated by a single element: a clockwise rotation by 120°,

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 \\ 10 & 9 & 19 & 4 & 12 & 18 & 5 & 17 & 11 & 13 & 2 & 7 & 1 & 8 & 6 & 3 & 14 & 15 & 16 \end{pmatrix}.$$

These symmetries allow us to generate additional state preparation circuits by permuting qubits while preserving the logical state.

A state preparation circuit $\mathcal{C}_1$ for the $|0\rangle_L$ with a CNOT depth of 5 and 27 CNOT gates is shown by the CNOT circuit in Fig. 3(a). Note that no CNOT gates act on qubits that do not have support on the same stabilizer generator. It follows immediately that CNOT gates on the qubits of the weight-4 stabilizers cannot propagate an error of weight greater than 2.

Furthermore, the CNOT gates on the weight-6 plaquettes only propagate errors of at most weight 2, namely,

$$\{X_1X_2, X_4X_5, X_{12}X_{14}, X_9X_{10}, X_4X_7, X_8X_{12}, X_{11}X_{13}\}.$$

Therefore, $\mathcal{E}_X(\mathcal{C}_1)$ is 2-distinct to itself. Setting

$$\mathcal{C} = \mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}_3 = \mathcal{C}_4$$

would therefore already fulfill requirements 1 and 2 of the condition (1).

To fully achieve fault tolerance, we must also ensure 2-distinctness for Z errors between the circuit pairs $\{\mathcal{C}_1, \mathcal{C}_2\}$ and $\{\mathcal{C}_3, \mathcal{C}_4\}$. There is (up to stabilizer) only one propagated Z error of weight 3, namely, $Z_4Z_5Z_{17}$. Applying a counterclockwise rotation σ acting on the qubits of $\mathcal{C}_1$ maps this error to $Z_4Z_{12}Z_{14}$, while applying a clockwise rotation σ^2 maps it to $Z_4Z_{12}Z_{11}$. Setting

$$\mathcal{C}_2 = \mathcal{C}_1, \quad \mathcal{C}_3 = \mathcal{C}_4 = \sigma(\mathcal{C}_1)$$

therefore yields four circuits that fulfill all the prerequisites of the condition (1).

B. ⟦17, 1, 5⟧ color code

Unlike the ⟦19, 1, 5⟧ code, the ⟦17, 1, 5⟧ code shown in Fig. 3(b) has only a trivial qubit symmetry. Thus, we cannot rely on permutations to obtain distinct fault sets and must instead modify the CNOT structure directly. Nevertheless, the same key arguments as with the ⟦19, 1, 5⟧ hold: Errors propagating on the weight-4 stabilizers can be ignored, and our focus shifts to the single weight-8 stabilizer. The state preparation circuit $\mathcal{C}_1$ in Fig. 3(b) has (up to stabilizers) only a single X error with weight greater than 2, namely, $X_5X_6X_7X_{11}$. All Z errors have a weight of at most 2, so $\mathcal{E}_Z(\mathcal{C}_1)$ is 2-distinct to itself.

We can construct a circuit with a different X fault set by altering the CNOT circuit on the weight-8 plaquette. One way to do this is to remove $\mathrm{CX}_{5,11}$ and add $\mathrm{CX}_{9,11}$ before $\mathrm{CX}_{9,16}$. This local modification yields a circuit $\mathcal{C}_2$ with single propagated X errors with weight greater than 2 of $\{X_5X_6X_7, X_{10}X_{11}X_{16}\}$. This set is 2-distinct to $\{X_5X_6X_7X_{11}\}$, so a Steane-type FTSP circuit can be constructed by setting $\mathcal{C}_3 = \mathcal{C}_1$ and $\mathcal{C}_4 = \mathcal{C}_2$.

V. AUTOMATED SYNTHESIS OF STEANE-TYPE FAULT-TOLERANT STATE PREPARATION CIRCUITS

Manual reasoning methods, which work well for smaller codes, as seen in Sec. IV, become increasingly difficult to apply as the code distance grows. For larger codes, fault sets can become enormous, and manually analyzing them quickly becomes infeasible. The same scaling affects the automated synthesis presented in this section: The cost of checking t-distinctness grows with the size of the involved fault sets, and the proportion of CNOT choices that preserve t-distinctness decreases as the constraints tighten. The method is therefore best suited to codes of moderate distance, where the fault sets are large enough to require automation but small enough that explicit enumeration remains feasible.

Section IV showed how a Steane-type FTSP circuit can be constructed by starting with a reference circuit and slightly altering it to generate another circuit with a t-distinct fault set. While this worked in the two examples shown, it is not generally possible to obtain a t-distinct circuit from a reference circuit only by local transformations. Our goal now is to automate this process by synthesizing a circuit whose fault set is t-distinct from a reference fault set by construction. To guarantee t-distinctness, we need control over the fault sets during synthesis so we can make informed choices about appropriate CNOT gates directly during circuit synthesis.

In this section we will describe such a fault-set-guided synthesis approach. Before we get to that, we briefly review how to construct state preparation circuits from a code check matrix.

A. Circuit synthesis via Gaussian elimination

A unitary $|0\rangle_L$ state preparation circuit for an $[\![n, k, d]\!]$ code with $H_X \in \mathbb{F}_2^{m_X \times n}$ and $H_Z \in \mathbb{F}_2^{m_Z \times n}$ is constructed by initializing m_X qubits in $|+\rangle$ and m_Z qubits in $|0\rangle$ followed by a CNOT circuit that transforms the check matrices of the initial product state to H_X and H_Z. The check matrices of the product state, $H_X^0 \in \mathbb{F}_2^{m_X \times n}$ and $H_Z^0 \in \mathbb{F}_2^{m_Z \times n}$, are matrices that are nonzero in exactly m_X and m_Z columns, respectively, such that the CSS condition holds. Applying a CNOT gate between qubits i and j to a CSS state adds column i to column j of the state's X check matrix and column j to column i in the state's Z check matrix.

The synthesis task asks for a sequence of CNOT gates that transform H_X to H_X^0 and H_Z to H_Z^0. This can be done via Gaussian elimination on the column space of the check matrix H_X. This elimination is also guaranteed to reduce H_Z as well by the CSS condition [18].

The state preparation circuit is then obtained by initializing the qubits corresponding to nonzero columns in H_X^0 to $|+\rangle$ and the others to $|0\rangle$ and running the elimination sequence in reverse, i.e., constructing the check matrices of the code from the check matrices of the product state. Since errors propagate backward through a circuit, building the circuit from the final layers first allows us to compute partial fault sets incrementally.

Naturally, the synthesized state preparation circuits should be minimal in some sense, either being low depth or using as few CNOT gates as possible. There is no known efficient algorithm for synthesizing the minimal circuit, so optimal synthesis approaches use techniques like satisfiability solving to find the optimal circuit [18]. However, efficient heuristics also exist based on greedy best-first search [18,29] or A* search [29].

B. Fault-set-guided synthesis

We are going to base our approach on the greedy best-first search proposed in Ref. [18]. This search evaluates the cost of every possible CNOT gate at each step by counting the number of nonzero elements in the resulting check matrix after applying it. The algorithm then greedily chooses the CNOT gate that eliminates the most nonzero entries. Multiple CNOT gates might have the same costs in this search, in which case a random CNOT gate is picked.

These degrees of freedom can be exploited to construct t-distinct circuits relative to a chosen reference circuit. As a first step, we synthesize such a reference circuit. Since there are no constraints on its fault set, we can obtain it using the heuristic from Ref. [18]. This circuit then serves as the baseline against which we construct the remaining circuits using fault-set-guided synthesis.

A crucial feature of the heuristic construction is that it builds the CNOT circuit in reverse, i.e., starting from the final layers and working backward. Since faults propagate in the opposite direction, the fault set of a circuit under construction can already be partially computed on the fly. Let $\mathcal{C}_k$ denote the intermediate circuit after k CNOT gates have been placed. If the intermediate CNOT circuit $\mathcal{C}_k$ has fault set $\mathcal{E}(\mathcal{C}_k)$, then the final circuit $\mathcal{C}$ will have a fault set $\mathcal{E}(\mathcal{C}_k) \subseteq \mathcal{E}(\mathcal{C})$. In particular, adding the gate CX_{ij} to $\mathcal{C}_k$ yields a circuit with fault sets $\mathcal{E}_X(\mathcal{C}_k) \cup \{\mathcal{C}_k X_i X_j \mathcal{C}_k^{-1}\}$ and $\mathcal{E}_Z(\mathcal{C}_k) \cup \{\mathcal{C}_k Z_i Z_j \mathcal{C}_k^{-1}\}$.

Crucially, any additional CNOT gates inserted later in the search can only contribute more faults; since the fault set is always computed with respect to the stabilizer group of

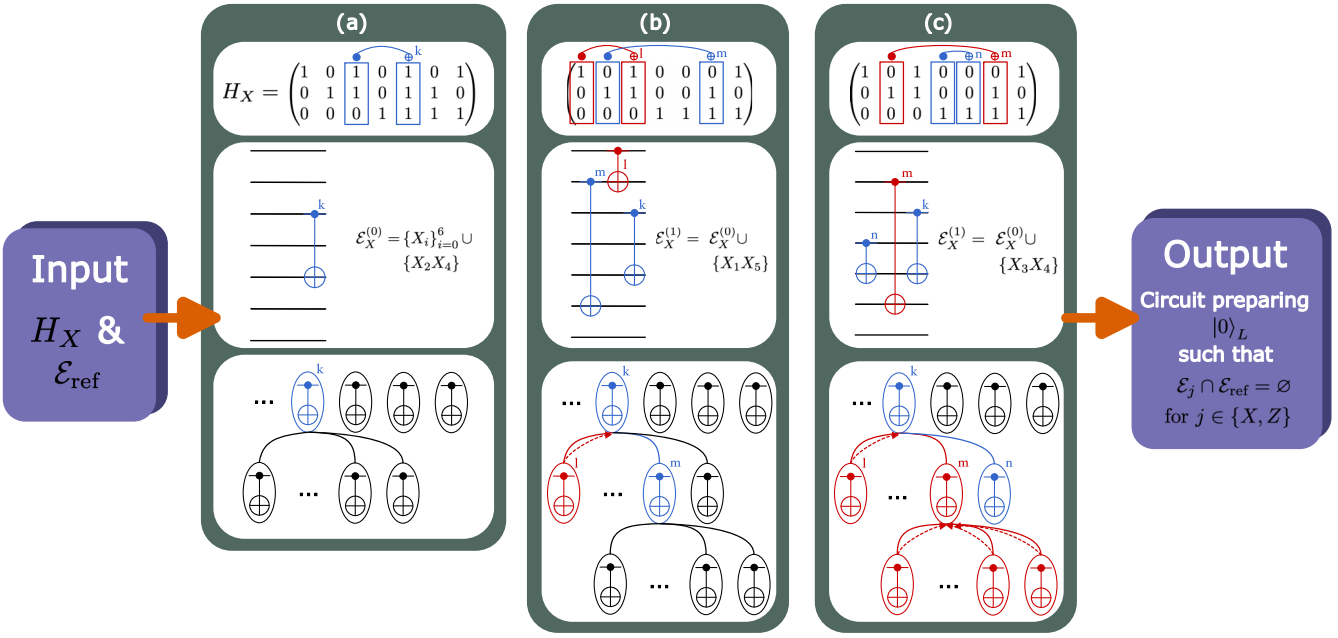


FIG. 4. Fault-set-guided synthesis of a state preparation circuit. Given the check matrix H_X and a reference fault set $\mathcal{E}_{\text{ref}}$, the algorithm constructs a circuit whose propagated fault set is t-distinct from $\mathcal{E}_{\text{ref}}$. (a)–(c) Three different scenarios in the search process are illustrated: the evolving check matrix (top), the partially constructed circuit assembled from right to left (middle), and the remaining candidate CNOT gates (bottom). Blue CNOT gates satisfy the t-distinctness constraint, while red CNOT gates violate it and are therefore pruned. (a) A valid CNOT gate is selected according to the cost function, and the fault set is updated. (b) A candidate CNOT gate violates t-distinctness and is discarded in favor of an alternative choice. (c) All candidates violate t-distinctness, forcing the algorithm to backtrack by removing previously placed CNOT gates.

the code and not of the intermediate state, no faults are ever removed from the intermediate fault set.

This on-the-fly construction of the evolving fault sets can be used to guide the search towards constructing t-distinct circuits. Given a reference fault set $\mathcal{E}_{\text{ref}}$, we can always evaluate whether adding a CNOT gate to the circuit under construction preserves t-distinctness of the constructed fault set and the reference fault set. At any stage of the search, we can filter the CNOT gates that would alter the fault set of the circuit under construction such that it would not be t-distinct from the reference circuit. Among the remaining candidate CNOT gates, we can select the one that optimizes the cost function. In practice, we first select a CNOT gate that locally optimizes the cost function and check whether it preserves t-distinctness. If it does not, the next best candidate is selected. This avoids many unnecessary and costly fault-tolerance checks up front. This is illustrated in Fig. 4(b).

This search may get stuck in regions of the search space where no CNOT gate placement maintains t-distinctness. In other words, all the CNOT gates we could place to prepare the desired logical state cause an overlap between the reference fault set and the current fault set of the circuit we are constructing at that moment. In this case, we have to backtrack by undoing previously chosen CNOT gates until the search can continue. This situation is illustrated in Fig. 4(c). The resulting algorithm is sketched in Algorithm 1.

Using this fault-set-guided-synthesis approach, we can construct the four circuits required for Steane-type FTSP.

(1) Construct $\mathcal{C}_1$ using some state preparation circuit synthesis approach (see, e.g., Refs. [18,29–31]).

(2) Construct $\mathcal{C}_2$ using fault-set-guided synthesis with respect to $\mathcal{E}_X(\mathcal{C}_1)$.

ALGORITHM 1. Fault-set-guided synthesis.

Input: Fault set $\mathcal{E}_{\text{ref}}$, fault-distance t, check matrix H_X, cost function h
Output: State preparation circuit with a t-distinct fault set to $\mathcal{E}_{\text{ref}}$

$\mathcal{C} \leftarrow$ empty circuits on n qubits;
$\mathcal{E} \leftarrow \{X_i \mid i \in [n]\} \cup \{Z_i \mid i \in [n]\}$;
$\mathcal{B} \leftarrow \emptyset$; /* blocked CNOT gates */
$H \leftarrow H_X$;
while *H is not reduced* **do**
  candidates $\leftarrow \text{argmin}_{\{\text{CX}_{ij} \mid i,j \in [n]\}} h(\text{CX}_{ij}, H)$;
  foreach $\text{CX}_{ij} \in$ candidates $\setminus \mathcal{B}$ **do**
    $\mathcal{E}' \leftarrow \mathcal{E} \cup \{\mathcal{C}(X_iX_j)\mathcal{C}^{-1}\} \cup \{\mathcal{C}(Z_iZ_j)\mathcal{C}^{-1}\}$;
    if *$\mathcal{E}'$ is t-distinct to $\mathcal{E}_{ref}$* **then**
      $\mathcal{C} \leftarrow \mathcal{C} \circ \text{CX}_{ij}$;
      $H[:, j] \leftarrow H[:, i] + H[:, j]$;
      $\mathcal{E} \leftarrow \mathcal{E}'$;
      Break;
  if *no valid CNOT gates can be added* **then**
    /* backtrack */
    Remove last gate cx_{last};
    Add cx_{last} to blocked list $\mathcal{B}$;
return $\mathcal{C}$

TABLE I. Comparison of fault-tolerant state preparation circuit synthesis methods. Here #CX denotes CNOT gate count; D is the CNOT depth, i.e., the number of layers of parallel CNOT gates (for Ref. [22] computed from the published circuits); p_l and r_A are the logical error rate and acceptance rate, respectively, for physical error rate $p = 10^{-3}$; and a blank entry indicates that no circuit is available. For each code, bold entries mark the best total CNOT count, depth, logical error rate, and acceptance rate among the compared methods.

	Ref. [18]				Ref. [22]				Fault-set-guided synthesis											
	Total		Error		Total		Error		$\mathcal{C}_1$		$\mathcal{C}_2$		$\mathcal{C}_3$		$\mathcal{C}_4$		Total		Error	
Code	#CX	D	p_l	r_A	#CX	D	p_l	r_A	#CX	D	#CX	D	#CX	D	#CX	D	#CX	D	p_l	r_A
$[\![17, 1, 5]\!]$	**71**	22	1.4×10^{-7}	0.92	74	17	4.4×10^{-7}	**0.92**	23	5	23	4	23	5	23	4	92	**7**	$\mathbf{8.0 \times 10^{-8}}$	0.79
$[\![19, 1, 5]\!]$	120	25	1.0×10^{-7}	**0.86**					27	4	27	4	27	6	27	5	**108**	**8**	$\mathbf{9.2 \times 10^{-8}}$	0.76
$[\![25, 1, 5]\!]$	127	31	7.4×10^{-8}	0.85	**92**	20	3.5×10^{-7}	**0.91**	28	5	28	5	28	5	28	5	112	**7**	$\mathbf{5.8 \times 10^{-8}}$	0.72
$[\![20, 2, 6]\!]$					**145**	51	1.6×10^{-6}	**0.84**	40	7	36	6	37	6	42	7	155	**9**	$\mathbf{8.6 \times 10^{-8}}$	0.72
$[\![31, 1, 7]\!]$	421	46	2.84×10^{-9}	0.56	**211**	55	4.1×10^{-8}	**0.77**	46	5	52	6	57	6	61	9	216	**11**	$\mathbf{1.1 \times 10^{-9}}$	0.62
$[\![37, 1, 7]\!]$									60	5	60	6	77	7	77	9	**274**	**11**	$\mathbf{1.0 \times 10^{-9}}$	**0.55**

(3) Construct $\mathcal{C}_3$ using fault-set-guided synthesis with respect to $\mathcal{E}_Z(\mathcal{C}_1) \cup \mathcal{E}_Z(\mathcal{C}_2)$.

(4) Construct $\mathcal{C}_4$ using fault-set-guided synthesis with respect to $\mathcal{E}_X(\mathcal{C}_3)$ and $\mathcal{E}_Z(\mathcal{C}_1) \cup \mathcal{E}_Z(\mathcal{C}_2)$.

The construction of $\mathcal{C}_4$ imposes the most constraints and in practice this leads to a significant increase in the number of backtracks. In larger codes, the space of valid state preparation circuits expands significantly, making fault-set-guided synthesis more difficult. Furthermore, higher distances demand costlier t-distinctness checks as the fault sets to be checked can get large.

To help the search and avoid getting stuck in difficult parts of the search space, we employ a few strategies.

(i) Avoid early conflicts by hard structural constraints. Besides the fault set of the reference circuit, we know the exact structure of the circuit itself. If the last two layers of the reference circuit contain the substructure $\mathrm{CX}_{ij}\mathrm{CX}_{jk}$, then we know that $X_iX_jX_k$ will be in the fault set (assuming it is not reducible by a stabilizer). We can therefore enforce that the same structure cannot be used for the circuit under construction without performing any fault-tolerance checks. It may also be beneficial to avoid structures that do not directly cause conflicts but may lead to them later. For example, it can be enforced that the last layer of CNOT gates of the circuit under construction cannot have any CNOT gates in common with the reference circuit. The idea is that such constraints increase the likelihood that errors propagate similarly in both circuits in the last layer.

(ii) Restart the search if the algorithm fails to make progress. A typical technique in many solvers [32] is to apply restarts, which help the search escape from unpromising parts of the search space where a choice far up the search tree causes the conflict. In our specific case, we can set a backtracking limit and restart the search once it is reached.

(iii) Apply random perturbations to the search. If multiple CNOT gates have the same costs, one can be picked at random. Additionally, a CNOT gate that has a (locally) higher cost might actually be a better choice for fault tolerance. We can increase the likelihood of selecting such CNOT gates by randomly selecting lower-cost CNOT gates with a probability parametrizing the search.

(iv) Apply row operations to the input check matrix. Since the check matrix represents a stabilizer group, row operations can be applied to it without altering the underlying group. The cost function depends on the specific shape of the check matrix; however, by extension, so does the search. We can therefore try to guide the search to favor synthesizing different CNOT circuits by changing the matrix representation. In practice, a good heuristic is to bring the matrix into reduced row-echelon form and synthesize the circuit from that.

Such heuristics help remove large parts of the search space early on.

VI. EVALUATION

We implemented the proposed fault-set-guided synthesis of Steane-type FTSP circuits in the open-source error-correction library MQT QECC [33], which is part of the Munich Quantum Toolkit [26]. Using this implementation, we generated Steane-type FTSP circuits for various $d = 5$, 6, and 7 CSS codes. In this section we evaluate these circuits in terms of CNOT gate count, depth, logical error rate, and acceptance probability using Monte Carlo simulations.

We compare the proposed method with Ref. [18], which ensures fault tolerance by measuring a small set of stabilizers after non-fault-tolerant unitary state preparation, and with Ref. [22], which augments the unitary preparation with flag gadgets that detect errors during initialization. In both cases, fault tolerance is achieved by postselecting on all measurement outcomes being $+1$.

A. Circuits

Table I shows the CNOT gate count and depth for the Steane-type FTSP circuits for various CSS codes generated using different fault-tolerant state preparation methods. The fault-set-guided-synthesis columns refer to circuits synthesized using the construction described in Sec. V. The proposed method synthesizes four copies of a state so as to illustrate the variance in circuit size and depth; we list the circuits $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ required for fault-tolerant Steane-type state preparation.

For the distance-5 codes, Table I shows that only minor adjustments to the reference circuits are required to achieve fault tolerance, with all designs exhibiting very similar CNOT gate counts and depths. This aligns with the analysis in Secs. IV B and IV A, which found that fault tolerance can be obtained

through small modifications to the original circuit structures. For instance, the optimized $[\![17, 1, 5]\!]$ circuit improves both metrics, while the $[\![19, 1, 5]\!]$ circuit incurs only a slight increase in depth.

For distance-7 codes, the synthesis task becomes more challenging due to the larger circuit sizes and fault sets. In particular, ensuring 3-distinctness for the Z-fault set of the fourth circuit is difficult, and a naive search often gets stuck. As described in Sec. V, we mitigate this using additional heuristics; for the $d = 7$ codes in Table I, we synthesize $\mathcal{C}_4$ starting from the reduced row-echelon form of the check matrix to guide the search.

Overall, the circuits generated by the proposed method compare favorably to Ref. [18], as fault-set-guided synthesis yields more compact circuits in every instance except for the $[\![17, 1, 5]\!]$ code, where Ref. [18] uses slightly fewer CNOT gates. In contrast, the so-called flag-at-origin construction of Ref. [22] consistently requires fewer CNOT gates but incurs a significantly higher depth overhead.

In the flag-at-origin scheme, circuits must be synthesized from the reduced row-echelon form of the check matrix and further augmented with flag gadgets, both of which increase depth. In our construction, the overall depth is determined by the maximum depth among the four state preparation circuits and by two additional layers of transversal CNOT gates for verification. As a result, the Steane-type FTSP protocol based on fault-set-guided synthesis achieves comparatively low depth.

B. Simulations

We simulated the synthesized circuits using the openly available stabilizer simulator Stim [34].

We employ a depolarizing noise model parametrized by a physical noise strength p.

(1) Two-qubit gates are followed by two-qubit depolarizing noise of strength p.

(2) Initializations and measurements flip with probability $\frac{2}{3}p$.

(3) Idle qubits in each circuit layer are subject to single-qubit depolarizing noise of strength $p/100$, and we assume gates can be executed in parallel with no crosstalk.

To estimate the logical X error rate, we simulate the construction in Fig. 1(b) for each code. Shots that indicate an error present on one of the three ancilla circuits are discarded. In kept shots, we perform one round of perfect error correction on the remaining state using a lookup-table decoder, measure the state in the standard basis, and check which eigenspace of the Z operator it ended up in. A logical -1 measurement indicates an error.

Getting an estimate on the Z error rate is trickier; a logical Z error leaves $|0\rangle_L$ unchanged. While $|0\rangle_L$ cannot directly be subject to logical Z errors, high-weight Z errors can still be an issue when using the prepared state as an ancilla in Steane-type error correction, where low-weight Z faults during preparation can cause a high-weight correction on the data state. Therefore, if preparation of the $|0\rangle_L$ is not strictly fault tolerant, such interactions can compromise the performance of any FTQC protocol that uses the prepared states.

We use a trick to estimate the logical error rate scaling for Z errors. To estimate Z errors on the $|0\rangle_L$, we can use it as an ancilla for Steane-type error correction. We initialize another logical qubit in the $|+\rangle_L$ state and subject it to single-qubit depolarizing noise of strength p. Afterward, we copy the Z error to the prepared $|0\rangle_L$ state with a transversal CNOT gate and measure out the state in the X basis. The syndrome is then decoded, and the correction applied to the $|+\rangle_L$ state. We can then estimate the logical Z error rate as before by measuring out the $|+\rangle_L$ state in the X basis. If the $|0\rangle_L$ state is prepared strictly fault tolerantly, then the correction applied to the $|+\rangle_L$ state due to the Z syndrome should not change the logical error rate scaling as events of weight t occur with probability $O(p^t)$, i.e., on the same order as in the single-qubit depolarizing channel.

The lookup table (LUT) decoder used in our simulations covers the entire syndrome space. Lookup table entries are populated by enumerating errors in order of increasing weight, with lower-weight errors taking precedence. For each syndrome, the first error encountered at its minimum weight is stored as the correction. This means that for syndromes corresponding to uniquely correctable errors [weight less than or equal to $\lfloor (d-1)/2 \rfloor$], the LUT applies the unique minimum-weight correction. For higher-weight syndromes, most notably weight-$(d/2)$ syndromes in the even-distance $[\![20, 2, 6]\!]$ code, where the code detects but cannot uniquely correct, the LUT applies one representative of the degenerate set of minimum-weight errors, with the specific choice determined by the iteration order. No additional postselection is applied beyond the four-ancilla check of the FTSP protocol, so events in which the LUT applies an incorrect representative contribute to the reported logical error rate.

Figure 5 shows the resulting logical error rates and acceptance rates of the simulated color code circuits for distances 5 and 7 generated by the fault-set-guided synthesis. Generally, we can see that error rates scale as $O(p^{\lceil d/2 \rceil})$ for the respective codes. The higher probability of logical Z errors is caused by the additional errors on the Steane-type error-correction gadget used to estimate the logical Z error rates.

Besides the color code simulations, Table I additionally summarizes the logical error rates and acceptance probabilities of the rotated surface code $[\![25, 1, 5]\!]$ and the $[\![20, 2, 6]\!]$ code for the proposed state preparation methods, alongside the approaches of Refs. [18, 22], evaluated at a physical error rate of 10^{-3}. Compared to Ref. [22], the proposed method exhibits consistently lower acceptance rates in simulation. Relative to Ref. [18], the acceptance rate is slightly reduced for distance-5 codes, while a clear improvement is observed for the distance-7 code. Although the proposed constructions exhibit very low circuit depths and a reduced number of CNOT gates compared to Ref. [18], as well as only a marginally higher CNOT gate count than Ref. [22], this does not translate into higher acceptance rates. Instead, the lower acceptance rates are an expected consequence of the verification strategy employed. Since all ancilla qubits are fully measured and errors on the data can propagate throughout the gadget, acceptance effectively postselects on the absence of errors across the entire circuit, up to acceptable cancellations. This stringent postselection naturally reduces the acceptance rate but simultaneously suppresses logical errors. As a result, the lower acceptance rate directly correlates with improved logical error performance. Indeed, the proposed method achieves

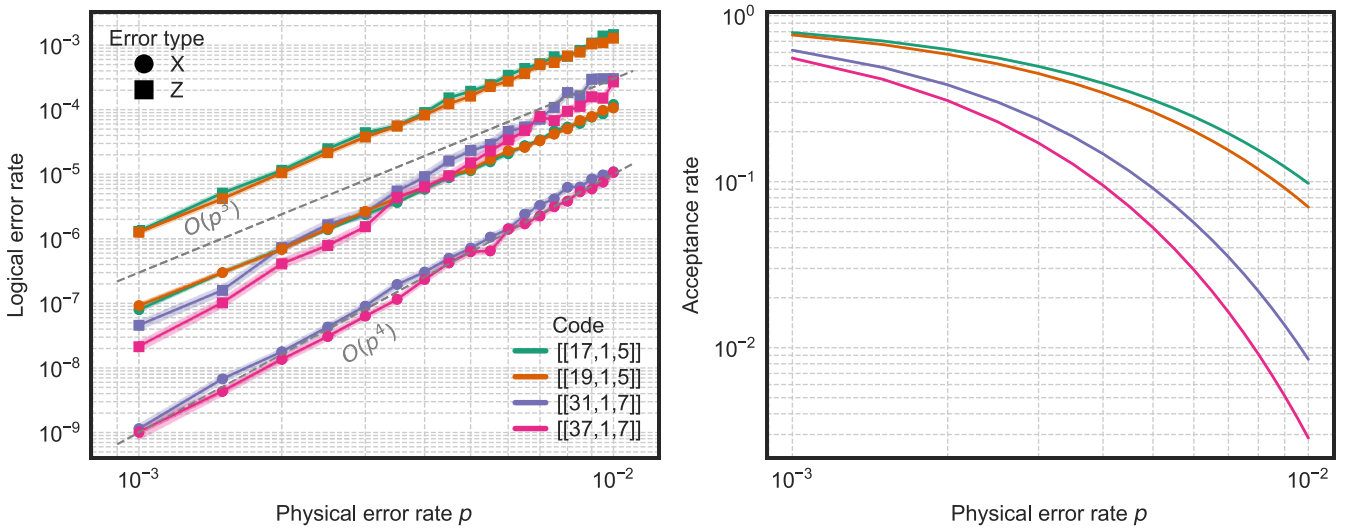


FIG. 5. Logical error and acceptance rates for Steane-type fault-tolerant state preparation circuits for the logical $|0\rangle_L$ state constructed with our methods for various $d = 5$ and 7 CSS codes under circuit-level noise using a LUT decoder.

logical error rates that are roughly an order of magnitude lower than those reported in Ref. [22] on average, with larger gaps for higher-distance codes, and performs similarly to Ref. [18] across all considered cases, with the exception of the ⟦20, 2, 6⟧ code. Overall, these results highlight a clear trade-off between acceptance rate and logical error suppression, under which the proposed constructions are preferable in settings where a lower acceptance rate is acceptable in exchange for significantly improved fault tolerance.

The synthesis methodology operates directly on the check matrices H_X and H_Z of the code and does not exploit any specific structural properties such as geometric locality, code symmetries, or the transversality of particular gates. The choice of color codes in Sec. IV reflects their suitability for illustrating the fault-set arguments, and the predominance of topological codes in Table I reflects their relevance to current experimental platforms; the method itself is code agnostic. The same procedure applies to other CSS code families relevant for fault-tolerant computation, including triorthogonal codes used for magic state distillation, subject to the same scaling considerations discussed in Sec. V.

VII. CONCLUSION

In this work we investigated the problem of Steane-type FTSP for arbitrary CSS codes in which a logical basis state of a code is prepared using four copies of a state, copying errors using transversal CNOT gates, and postselecting on the trivial measurement result. We showed how this can be achieved by altering the circuit structure of the CNOT circuit used for state preparation, either by permuting commuting CNOT gates or by using on-the-fly construction of preparation circuits utilizing a backtracking approach. Furthermore, we implemented the proposed methods and provided an open-source toolkit for constructing and evaluating Steane-type FTSP circuits.

While we have shown the efficacy of the proposed fault-set-guided construction on selected $d = 5$ and 7 codes, the need for explicit construction of large fault sets for higher-distance codes poses an obstacle to the scalability of this approach. Future work could aim to alleviate or optimize this by trying a combination of automated approaches developed recently [18,22]. It might be possible to use the error detection of these constructions to reduce the size of the fault sets involved.

This work provides an important step in the compilation of fault-tolerant quantum programs. On the one hand, the states generated by this repeat-until-success protocol might be interesting for near-term demonstrations of fault-tolerant quantum computing with slow measurement and no real-time classical feedback. On the other hand, high-quality states prepared using the optimized circuits might be of interest for fault-tolerant quantum computing with Steane-type error correction.

ACKNOWLEDGMENTS

The authors would like to thank Sascha Heußen for initial discussions on generalizing Steane-type state preparation of the GOLAY code, Ludwig Schmid for insightful discussions and comments, and Lucas Berent for helpful comments on the initial version of this paper. The authors acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 101001318), from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101114305 ("MILLENION-SGA1" EU Project), and from the Munich Quantum Valley (MQV, K5 and K7), which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. Furthermore, this work was supported by the BMFTR under Grant No. 13N17298 (SYNQ) and the Deutsche Forschungsgemeinschaft (German Research Foundation) under Grants No. 563402549 and No. 563436708.

DATA AVAILABILITY

The data that support the findings of this article are openly available [35]. The implementation of the proposed methods is available as part of the open-source library MQT QECC [33].

[1] A.Y. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. (N.Y.) **303**, 2 (2003).

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 2nd ed. (Cambridge University Press, Cambridge, 2010).

[3] D. Gottesman, Theory of fault-tolerant quantum computation, Phys. Rev. A **57**, 127 (1998).

[4] P. W. Shor, Fault-tolerant quantum computation, in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science, Burlington, 1996* (IEEE, Piscataway, NJ, 1996), pp. 56–65.

[5] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A **54**, 1098 (1996).

[6] A. M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett. **77**, 793 (1996).

[7] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, Caltech, 1997.

[8] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, Phys. Rev. Lett. **78**, 2252 (1997).

[9] L. Postler, F. Butt, I. Pogorelov, C. D. Marciniak, S. Heußen, R. Blatt, P. Schindler, M. Rispler, M. Müller, and T. Monz, Demonstration of fault-tolerant Steane quantum error correction, PRX Quantum **5**, 030326 (2024).

[10] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, J. G. Bohnet, N. C. Brown, N. Q. Burdick, W. C. Burton, S. L. Campbell, J. P. Campora, C. Carron, J. Chambers, J. W. Chan, Y. H. Chen, *et al.*, A race-track trapped-ion quantum processor, Phys. Rev. X **13**, 041052 (2023).

[11] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, Demonstration of the trapped-ion quantum CCD computer architecture, Nature (London) **592**, 209 (2021).

[12] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, Parallel entangling operations on a universal ion-trap quantum computer, Nature (London) **572**, 368 (2019).

[13] M. Saffman, Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges, J. Phys. B **49**, 202001 (2016).

[14] M. Morgado and S. Whitlock, Quantum simulation and computing with Rydberg-interacting qubits, AVS Quantum Sci. **3**, 023501 (2021).

[15] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, H. Pichler, and M. D. Lukin, Parallel implementation of high-fidelity multiqubit gates with neutral atoms, Phys. Rev. Lett. **123**, 170503 (2019).

[16] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, *et al.*, Logical quantum processor based on reconfigurable atom arrays, Nature (London) **626**, 58 (2024).

[17] H. Goto, Minimizing resource overheads for fault-tolerant preparation of encoded states of the Steane code, Sci. Rep. **6**, 19578 (2016).

[18] T. Peham, L. Schmid, L. Berent, M. Müller, and R. Wille, Automated synthesis of fault-tolerant state preparation circuits for quantum error-correction codes, PRX Quantum **6**, 020330 (2025).

[19] B. W. Reichardt, A. Paetznick, D. Aasen, I. Basov, J. M. Bello-Rivas, P. Bonderson, R. Chao, W. van Dam, M. B. Hastings, A. Paz, M. P. da Silva, A. Sundaram, K. M. Svore, A. Vaschillo, Z. Wang, M. Zanner, W. B. Cairncross, C.-A. Chen, D. Crow, H. Kim, *et al.*, Fault-tolerant quantum computation with a neutral atom processor, arXiv:2411.11822.

[20] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).

[21] R. Chao and B. W. Reichardt, Flag fault-tolerant error correction for any stabilizer code, PRX Quantum **1**, 010302 (2020).

[22] D. Forlivesi and D. Amaro, Flag at origin: A modular fault-tolerant preparation for CSS codes, arXiv:2508.14200.

[23] A. Paetznick and B. W. Reichardt, Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit Golay code, Quantum Inf. Comput. **12**, 1034 (2012).

[24] A. Gong and J. M. Renes, Computation with quantum Reed-Muller codes and their mapping onto 2D atom arrays, arXiv:2410.23263.

[25] T. Peham, E. Weilandt, and R. Wille, Optimizing fault-tolerant cat state preparation, Phys. Rev. A **113**, 062426 (2026).

[26] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schönberger, Y. Stade, and L. Burgholzer, The MQT handbook: A summary of design automation tools and software for quantum computing, in *Proceedings of the 2024 IEEE International Conference on Quantum Software (QSW)* (IEEE, Shenzhen, 2024), pp. 1–8.

[27] A. M. Steane, Simple quantum error-correcting codes, Phys. Rev. A **54**, 4741 (1996).

[28] H. Bombin and M. A. Martin-Delgado, Topological quantum distillation, Phys. Rev. Lett. **97**, 180501 (2006).

[29] M. Webster, S. Koutsioumpas, and D. E. Browne, Heuristic and optimal synthesis of CNOT and Clifford circuits, Quantum **10**, 2212 (2026).

[30] A. M. Steane, Fast fault-tolerant filtering of quantum codewords, arXiv:quant-ph/0202036.

[31] R. Zen, J. Olle, L. Colmenarez, M. Puviani, M. Müller, and F. Marquardt, Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning, Phys. Rev. X **15**, 041012 (2025).

[32] A. Biere, M. Heule, H. van Maaren, and T. Walsh, *Handbook of Satisfiability* (IOS Press, Amsterdam, 2009).

[33] L. Berent, T. Peham, L. Burgholzer, L. Schmid, P.-J. H. S. Derks, T. Grurl, and C. Pichler, MQT QECC, GitHub, 2025, https://github.com/munich-quantum-toolkit/qecc.

[34] C. Gidney, Stim: A fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[35] E. Weilandt, T. Peham, and R. Wille, Data for "Synthesis of fault-tolerant state preparation circuits using Steane-type error detection", Zenodo, 2026, doi:10.5281/zenodo.23183006.