Facilitating practical fault-tolerant quantum computing based on color codes

Jiaxuan Zhang,[1,2,3] Yu-Chun Wu,[1,2,3,4,*] and Guo-Ping Guo[1,2,3,4,5]

[1] *Key Laboratory of Quantum Information, Chinese Academy of Sciences, School of Physics,*
University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China
[2] *CAS Center For Excellence in Quantum Information and Quantum Physics,*
University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China
[3] *Hefei National Laboratory, University of Science and Technology of China, Hefei 230088, People's Republic of China*
[4] *Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui 230088, People's Republic of China*
[5] *Origin Quantum Computing Hefei, Anhui 230026, People's Republic of China*

(Received 20 September 2023; accepted 3 July 2024; published 19 July 2024)

Color codes are a promising topological code for fault-tolerant quantum computing. Insufficient research on color codes has delayed their practical application. In this work, we address several key issues to facilitate practical fault-tolerant quantum computing based on color codes. First, by introducing decoding graphs with error-rate-related weights, we obtain the threshold of 0.47% of the 6.6.6 triangular color code under the standard circuit-level noise model, narrowing the gap to that of the surface code. Second, our work first investigates the circuit-level decoding of color code lattice surgery, then gives an efficient decoding algorithm, which is crucial to perform logical operations in a quantum computer with two-dimensional architectures. Last, a state injection protocol of the triangular color code is proposed, reducing the output magic state error rate in one round of 15 to 1 distillation by two orders of magnitude compared to a previous rough protocol. We also prove that our protocol offers the lowest logical error rates for state injection among all possible CSS codes.

DOI: 10.1103/PhysRevResearch.6.033086

I. INTRODUCTION

Quantum computation provides a potential way to solve classically intractable problems, such as integer factorization [1] and simulation of large quantum systems [2,3]. To enable practical large-scale quantum computation, quantum error correction (QEC) is a crucial technique that protects information against the noise by encoding physical qubits into logical qubits of a certain type of QEC code [4–6].

Much of the current work focuses on a subset of QEC codes named the stabilizer code [7]. As a type of stabilizer code, topological codes [8,9], including color codes [10,11] and surface codes [12], draw extra attention since they are compatible with the real hardware limited by the local constraints in a two-dimensional (2D) architecture. Compared to surface codes, the advantage of color codes is that they encode a logical qubit with fewer physical qubits and can implement logical Clifford gates transversally.

An important indicator of the performance of a QEC code is the threshold. However, it had long been believed that the circuit-level threshold of color codes is about an order of magnitude lower than that of surface codes, which is a significant reason why color codes fall behind in the competition with surface codes [13–16]. In the first part of our work, by improving the decoding algorithm, we obtain the threshold of 0.47% for the 6.6.6 triangular color code [17], basically the same order of magnitude as the threshold of surface codes of ∼0.7–0.9 using an efficient decoder [15,16].

When performing logical operations on 2D hardware, lattice surgery [13,18–21] is currently the mainstream scheme since it retains locality constraints and offers lower overhead compared to braiding schemes [22]. Lattice surgery implements logical gates, including Clifford gates and non-Clifford gates, through fault-tolerant measurements of multibody logical Pauli operators. Recent work shows that lattice surgery of color codes can further reduce the time cost by measuring an arbitrary pair of commuting logical Pauli operators in parallel [19], which offers additional evidence of the advantage of color codes. Moreover, in the lattice surgery of surface codes, measurements involving Pauli Y operators typically need to introduce twist defects, which increase the requirement of device connectivity and resource costs [23,24]. In spite of Ref. [25] that proposes a twist-free lattice surgery scheme, it still requires additional space and time overhead, while in color code lattice surgery, the difficulties of measuring Y-type logical Pauli operators are almost nonexistent since they can transform into X- or Z-type Pauli operators through transversal single-qubit Clifford gates. Although color code lattice surgery is theoretically feasible and may have advantages, a decoding strategy at the circuit level is still lacking. Our work investigates this process and proposes an efficient circuit-level decoding algorithm for a class of color code lattice surgery.

[*] Contact author: wuyuchun@ustc.edu.cn

Magic states, as the ancilla states for performing non-Clifford gates [26], represent another widely studied research focus in fault-tolerant quantum computing. Obtaining high-fidelity magic states requires an expensive process called magic state distillation [27–29]. The first step of magic state distillation is injecting a physical magic state into a logical state. The quality of the initial logical magic states from state injection will remarkably affect the error rate of the output state after distillation. Previous research has demonstrated that magic states from a surface code state injection protocol have a better fidelity than the operations used to construct them [30–32]. In contrast, well-designed state injection protocols for color codes and error rate analyses of them are still quite limited. This work proposes a state injection protocol of the color code. Compared to a rough protocol used in Ref. [33], our protocol reduces the output magic state error rate in one round of 15 to 1 distillation by two orders of magnitude.

Overall, there are three main results in this paper. First, a decoding algorithm is proposed, achieving a threshold of 0.47% for 6.6.6 triangular color codes. Based on the projection decoder [33,34], we construct decoding graphs by an automated procedure and introduce error-rate-related weights to implement more accurate matching by the minimum weight perfect matching (MWPM) algorithm [35,36]. Our algorithm significantly improves the decoding accuracy of the previous 6.6.6 triangular color code decoder [33] under circuit-level noise. The numerical results show that the threshold under the circuit-level noise model is around 0.47%, which, to the best of our knowledge, is the highest threshold among all types of 2D color codes without circuit optimization. We also note that in Ref. [37], a higher threshold for color codes is demonstrated. However, this is primarily due to their optimization of the stabilizer measurement circuits rather than the decoder itself, and thus cannot be directly compared to our results. Second, we demonstrate the decoding process of lattice surgery between two logical qubits under a circuit-level noise model. This algorithm is applicable to lattice surgery schemes on color code with color boundaries [38]. We simulate an example in which logical operators $X_L \otimes X_L$ and $Z_L \otimes Z_L$ are measured in parallel and find that the spacelike effective code distance is slightly less than that of the timelike one, and the spacelike error is the dominated logical error in color code lattice surgery. Lastly, we investigate state injection of the color code and give a protocol based on postselection. The performance of our protocol is superior to the existing state injection protocols of surface codes in terms of logical error rates, postselection success rates, and process complexity. Furthermore, it has been proven that the logical error rate of this protocol is lowest compared to any state injection protocol among all possible CSS codes [39,40]. We also discuss how to design a proper postselection scheme by the correlation coefficients of syndrome changes and the occurrence of logical errors.

The remainder of this paper is organized as follows. Section II reviews some basic but important concepts in fault-tolerant quantum computation and specifies notations used in this paper. Section III introduces the improved color code decoding strategy and presents the numerical results of the threshold. In Sec. IV, we describe the circuit-level decoding of color code lattice surgery and discuss its applicability. Then, in Sec. V, the state injection protocol of color codes is proposed, where we use a theorem to summarize its optimality. Finally, the conclusion and outlook are presented in Sec. VI.

II. PRELIMINARIES

In this section, we briefly review some important background materials for 2D color codes, which are referred to throughout the paper. Section II A introduces the basic definitions and notations about 6.6.6 triangular color codes. In Sec. II B, we discuss the Pauli-based computation and how the lattice surgery performs in the 2D color codes. In Sec. II C, we review another key problem of the fault-tolerant quantum computation—magic state distillation. Lastly, the details of the circuit-level noise model are presented in Sec. II D, which is the premise of the following discussions and numerical results.

A. Triangular color codes and notations

2D color codes are topological QEC codes constructed in a three-colorable and trivalent lattice. In this paper, we focus on the hexagonal color code with triangular boundaries (referred to as the triangular color code below). The triangular color code is defined in the hexagonal lattice $\mathcal{L}$ with three boundaries. Let us denote the sets of vertices, edges, and faces of a lattice (or a graph) $\mathcal{L}$ by $V(\mathcal{L})$, $E(\mathcal{L})$, and $F(\mathcal{L})$, respectively. As illustrated in Fig. 1(a), each data qubit of the triangular color code is placed in a vertex $v \in V(\mathcal{L})$, and each face $f \in F(\mathcal{L})$ corresponds to two stabilizer generators $S_f^X = \otimes_{v\in f} X_v$ and $S_f^Z = \otimes_{v\in f} Z_v$, respectively. The logical code space is the simultaneous eigenspace of all the stabilizers and the measurement results of all stabilizer generators form a binary string called syndrome. The triangular color code encodes one logical qubit, a representative logical Pauli operator which can be defined as the tensor product of Pauli operators supported by a string in the boundary. The minimum weight of all logical X or Z operators is defined as the code distance, denoted by d. Note that color codes are Calderbank-Shor-Steane stabilizer codes and self-dual, i.e., the supports of S_f^X and S_f^Z are the same. Hence, the X or Z errors can be corrected separately in an analogous manner.

Given the primal lattice $\mathcal{L}$ of the color code, it is convenient to discuss the decoding problem on its dual lattice $\mathcal{L}^*$. In the dual lattice $\mathcal{L}^*$, each face corresponds to a data qubit and each vertex corresponds to two stabilizer generators S_f^X and S_f^Z, except for the three boundary vertices v_R, v_G, and v_B [see Fig. 1(b)]. We use $F_1 \subseteq F(\mathcal{L}^*)$ to denote the set of faces identified with qubits affected by X (or Z) errors and $\sigma \subseteq V(\mathcal{L}^*)$ to denote the vertex set in which the identified stabilizer generators anticommutes with the Pauli error supported by F_1.

We also need to define three restricted lattices $\mathcal{L}_C^*$, where $C \in \{RG, RB, GB\}$. The restricted lattice, say $\mathcal{L}_{RG}^*$, is obtained from $\mathcal{L}^*$ by removing all blue vertices of $\mathcal{L}^*$ as well as all the edge and faces incident to the removed vertices.

Moreover, for color code decoding under the circuit-level noise model, 3D lattices or graphs are usually employed, where the vertical direction represents the time layers, and each time layer corresponds to a 2D lattice. We use

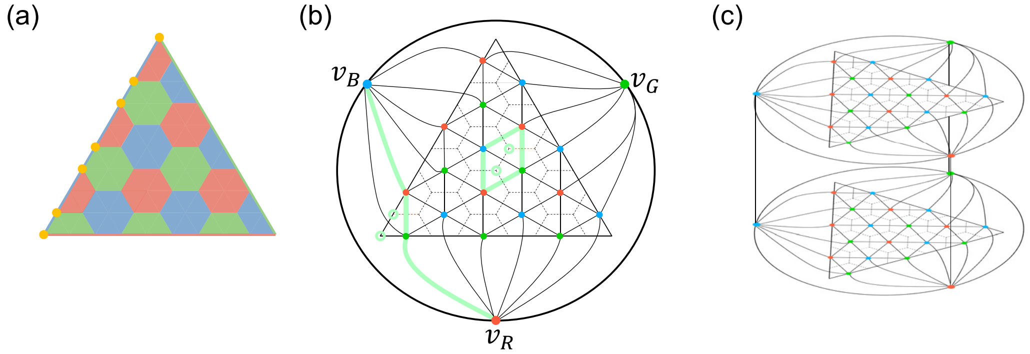


FIG. 1. (a) Primal lattice $\mathcal{L}$ of the triangular color code. Each hexagonal or half-hexagonal face corresponds to two stabilizers which are the tensor products of X and Z supported by the qubits located in the vertices of the face, respectively. A representative of the X (or Z) logical operator is defined as tensor products of X (or Z) supported by the qubits located in a string (colored by yellow). (b) Dual lattice $\mathcal{L}^*$ of the triangular color code with $d = 7$. The green lines and circles are two examples of paths and the corresponding corrections in the decoding. (c) 3D lattice $\mathcal{L}^* \times t$ (here, $t = 2$). Each layer of $\mathcal{L}^* \times t$ is the 2D dual lattice $\mathcal{L}^*$. The corresponding vertices of adjacent layers are connected vertically. To avoid clutter, we only show the vertical connections of three boundary vertices.

v_i^t to denote the vertex in a 3D lattice or graph, where $t \in \mathbb{Z}^+$ is the number of layers in which v_i^t is located, and v_i is the projection of v_i^t on the 2D lattice. One type of 3D lattice that we frequently use is $\mathcal{L}^* \times n$, which is defined by $V(\mathcal{L}^* \times n) = \{v_i^t | v_i \in V(\mathcal{L}^*), t = 1, 2, \dots, n\}$ and $E(\mathcal{L}^* \times n) = \{(v_i^t, v_j^t) | (v_i, v_j) \in E(\mathcal{L}), t = 1, 2, \dots, n\} \cup \{(v_i^t, v_i^{t+1}) | v_i \in V(\mathcal{L}^*), t = 1, 2, \dots, n-1\}$, where $\mathcal{L}^*$ is a 2D lattice and n is a positive integer. In particular, we say the two edges $e_1 = (v_1^{t_1}, v_2^{t_2})$ and $e_2 = (v_3^{t_3}, v_4^{t_4})$ are parallel in a 3D lattice if $v_1 = v_3$, $v_2 = v_4$ and $t_3 - t_1 = t_4 - t_2$, notated by $e_1 // e_2$.

Lastly, it is also required to define the path s in a lattice or graph $\mathcal{G}$ to be a sequence of vertices $s = [v_1^{t_1}, v_2^{t_2}, \dots, v_n^{t_n}]$, where $(v_k^{t_k}, v_{k+1}^{t_{k+1}}) \in E(G)$ for any $k \in 1, 2, \dots, n-1$. The reverse of s is defined as $\bar{s} = [v_n^{t_n}, v_{n-1}^{t_{n-1}}, \dots, v_1^{t_1}]$. We refer to $s^{(k)}$ as the kth vertex in path s and refer to the first and last vertices as the endpoints of s. If an edge e satisfies $e = (s^{(k)}, s^{(k+1)})$, we say edge e is on the path s. If the two endpoints of path s are the same, we say s is enclosed. In addition, two functions acting on the paths are defined as follows. First, if s_1 and s_2 have the same endpoint, we say s_1, s_2 can be concatenated and $s = s_1 + s_2$ is the concatenation of s_1 and s_2, where s is the sequence of $n + m - 1$ vertices satisfying $[s^{(1)}, s^{(2)}, \dots, s^{(n)}] = s_1$ or $\bar{s}_1$, and $[s^{(n)}, s^{(n+1)}, \dots, s^{(n+m-1)}] = s_2$ or $\bar{s}_2$. Second, suppose $s = [v_1^{t_1}, v_2^{t_2}, \dots, v_n^{t_n}]$ is a path in a 3D lattice $\mathcal{L}^* \times n$; then the projection of s is defined as $[v_1, v_2, \dots, v_n]$, denoted by proj(s), which is a path in the 2D lattice $\mathcal{L}^*$. Here the projection is modulo 2, which means that if an even number of edges on s are projected into the same edge, it is equivalent to deleting this edge in the projection.

B. Color code lattice surgery and Pauli-based computation

Lattice surgery [13,18–21] is a measurement-based scheme allowing for efficient implementation of universal gate sets by fault-tolerant logical multibody Pauli operator measurements. It is especially suitable for 2D topological quantum codes, such as color codes or surface codes, as it only requires 2D qubit layout and local interactions.

For triangular color codes, Fig. 2(a) shows an example of lattice surgery for measuring $X_L \otimes X_L$ of two logical qubits. Two triangular logical qubits are merged by an ancillary lattice in the middle, where data qubits are initialized to Bell pairs. Then the QEC cycles are performed to measure the stabilizers given in Fig. 2(c). The outcome of the $X_L \otimes X_L$ measurement is determined by the product of some stabilizer measurement outcomes, based on $X_L \otimes X_L = S^* B^*$, where S^* is the product of X-type stabilizers labeled by a star and B^* is the product of the stabilizers $X \otimes X$ of yellow Bell states in Fig. 2(a). After repeating d rounds of QEC circuits for fault tolerance, two logical qubits are split by measuring their own stabilizers. Based on $X_L \otimes X_L$ measurement and transversal logical Clifford gates of color codes, one can realize arbitrary two-qubit logical Pauli operator measurements by lattice surgery.

Different from surface codes, lattice surgery of color codes allows arbitrary pairs of commuting logical Pauli measurements in parallel, while the space cost remains the same [19]. For example, keeping the qubit layout in Fig. 2(a), parallel measurements of $X_L \otimes X_L$ and $Z_L \otimes Z_L$ can be achieved as long as the stabilizers that connect the logical qubits and ancillary lattice are replaced with the stabilizers shown in Fig. 2(b). References [19,38] give more general examples of color code lattice surgery with different types of weight-4 and weight-8 stabilizers at the boundaries between regions.

Through lattice surgery, one can perform gate-based computation where a universal gate set is realized by Pauli operator measurements and ancilla qubits [22]. However, with the capacity of measuring arbitrary Pauli operators, a natural way to achieve universal quantum computing is Pauli-based computation (PBC) [25,41,42].

In the PBC model, a gate-based quantum circuit is equivalent to a series of Pauli operator measurements on initial states $|0^{\otimes n}\rangle$ and ancilla magic states $|m^{\otimes k}\rangle$. To illustrate this, let us start with a Clifford+T circuit in Fig. 3, which is well known to form a universal gate set. By commuting all T gates to the foreside of the circuit, the Clifford+T circuit is replaced with a series of $\frac{\pi}{8}$ Pauli rotations $P(\frac{\pi}{8}) = e^{-i\pi P/8}$, followed by Clifford gates C' and Z-basis measurements. Then, a $\pi/8$ Pauli rotation can be realized by Pauli operator measurement M_i and Clifford gates with an ancilla magic state [see Fig. 4(a)].

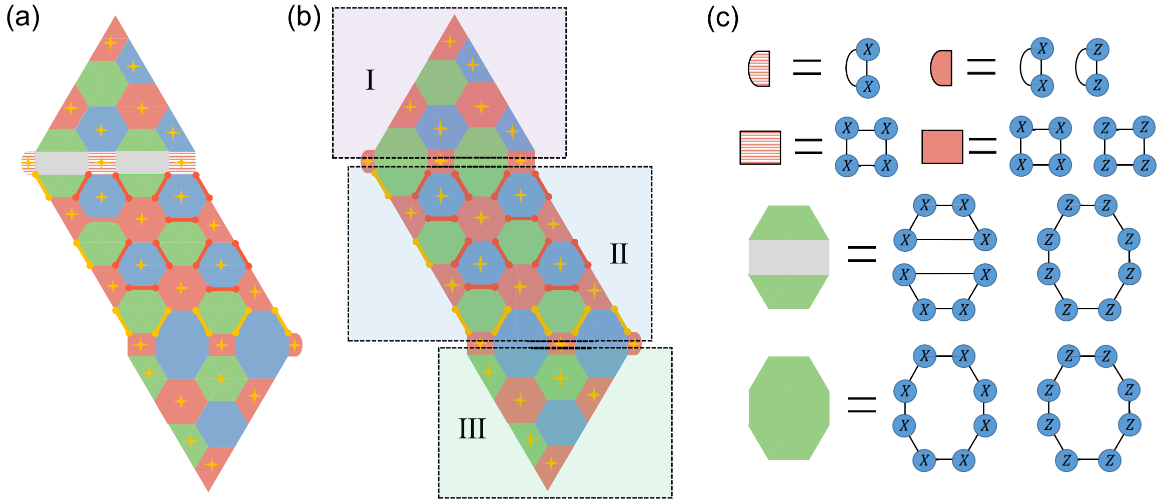


FIG. 2. (a) Lattice surgery of triangular color codes for measuring $X_L \otimes X_L$. The qubits in the middle part are initialized to Bell states (red and yellow pairs). Then the result is the product of the measurement outcomes of starred X-type stabilizers. (b) Lattice surgery of triangular color codes for measuring $X_L \otimes X_L$ and $Z_L \otimes Z_L$ in parallel. The results are the product of the measurement outcomes of starred X-type (or Z-type) stabilizers. The lattice is divided into three regions, comprising the upper and lower logical qubit regions and the middle auxiliary lattice region (marked by three different colors). At the boundaries between two regions, there exist stabilizer generators with weights of 8 and 4 (or 2). (c) Specific forms of the stabilizer generators at the boundaries between regions.

Lastly, we commute all Clifford gate and Z-basis measurements, which transforms Z-basis measurements to multibody Pauli operator measurements. Therefore, we have proven that the PBC model is equivalent to a universal gate-based computation model.

Note that the outcome of measurement M_i may affect subsequent measurements since there is a Clifford gate $P(\frac{\pi}{4})$ controlled by measurement outcome when we perform $P(\frac{\pi}{8})$ gate. Hence, the execution time of the PBC circuit is linearly related to the layers of $P(\frac{\pi}{8})$ gates where $P(\frac{\pi}{8})$ gates commute in the same layer. Some research has focused on reducing the time overhead. For instance, the time-optimal scheme [21,43] allows parallelism of the measurement layers by using a large number of ancilla qubits, and the temporally encoded lattice surgery scheme [25] effectively reduces the time per measurement.

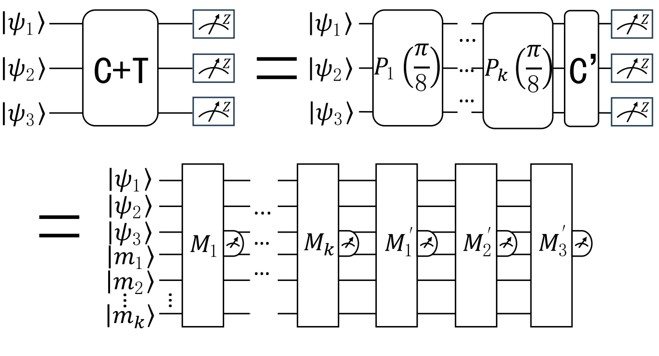
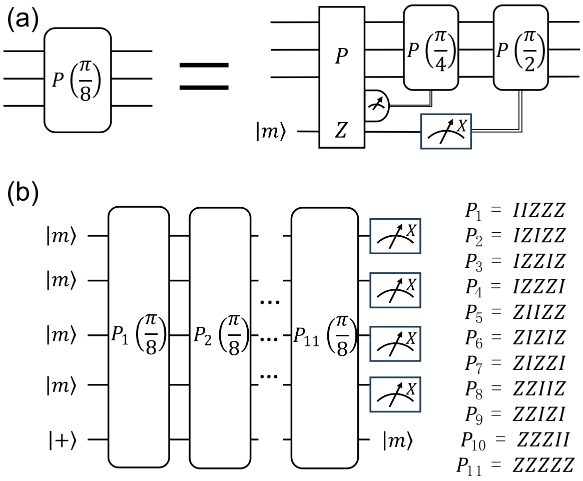


FIG. 3. Equivalence of the gate-based quantum computation model and the PBC model. Any computation can be expressed as a Clifford+T circuit. First we commute all the Clifford gates to the right of the circuit and then perform all $P(\frac{\pi}{8})$ gates by introducing magic states $|m_i\rangle$ and multibody Pauli operator measurements M_i. Lastly, the Clifford gates can be absorbed by the final measurements.

C. Magic state distillation

In general, a practical quantum computer requires a universal gate set. According to the Eastin-Knill theorem [44], no QEC code admits a universal and transversal logical gate set. The logical Clifford gates in the topological quantum codes are typically easier to implement by lattice surgery or braiding operations [15,45]. However, performing non-Clifford logical gates such as a $\pi/8$ Pauli rotation gate $P(\frac{\pi}{8})$ requires an extra resource state $|m\rangle = |0\rangle + e^{i\pi/4}|1\rangle$, called the magic state.

Utilizing a magic state as the ancilla qubit, one can realize a logical gate $P(\frac{\pi}{8})$ by the circuits in Fig. 4(a) using multibody Pauli operator measurements and classically controlled Clifford gates. Suppose the Pauli operator measurements


(a)
$P\left(\frac{\pi}{8}\right)$
=
P
Z
$P\left(\frac{\pi}{4}\right)$
$P\left(\frac{\pi}{2}\right)$
$|m\rangle$
X
(b)
$|m\rangle$
$|+\rangle$
$P_1\left(\frac{\pi}{8}\right)$
$P_2\left(\frac{\pi}{8}\right)$
$P_{11}\left(\frac{\pi}{8}\right)$

$P_1 = IIZZZ$
$P_2 = IZIZZ$
$P_3 = IZZIZ$
$P_4 = IZZZI$
$P_5 = ZIIZZ$
$P_6 = ZIZIZ$
$P_7 = ZIZZI$
$P_8 = ZZIIZ$
$P_9 = ZZIZI$
$P_{10} = ZZZII$
$P_{11} = ZZZZZ$

FIG. 4. (a) $\pi/8$ rotation gate implemented by consuming a magic state. (b) Circuit of the 15 to 1 magic state distillation protocol. Four magic states are consumed in the initial states and 11 magic states are consumed to implement $P(\frac{\pi}{8})$ gate (total of 15).

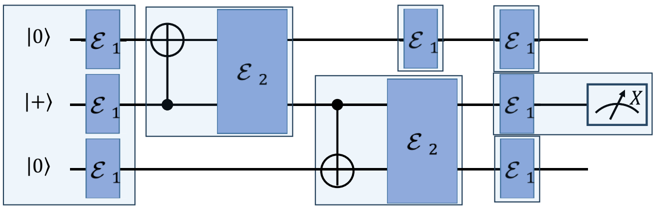


FIG. 5. An example of the noisy circuit to prepare a Bell state. We label the noise channels of the initialization errors, two-qubit gate errors, idling errors, and measurement errors separately.

and Clifford gates are fault-tolerantly implemented by lattice surgery, and the fidelity of the logical gate $P(\frac{\pi}{8})$ is mainly determined by the fidelity of $|m\rangle$. High-fidelity magic states can be produced by several copies of noisier magic states, which is called magic state distillation. Magic state distillation is often considered to be the most costly part of fault-tolerant quantum computing, even though there are an increasing number of proposed efficient protocols [29,46–49].

Here, we briefly review the most well-known magic state distillation protocol, i.e., the 15 to 1 protocol, as an example [21]. This protocol is based on the 15-qubit Reed-Muller code, which is the smallest QEC code with the transversal T gate [50]. The distillation circuit starts with four magic states $|m\rangle$ and a $|+\rangle$ state, followed by $\pi/8$ Pauli rotation gates and X-basis measurements, as shown in Fig. 4(b). Since one magic state is consumed in each $\pi/8$ Pauli rotation gate implementation, a total of 15 magic states are consumed to distill one high-fidelity magic state. The errors of, at most, two of 15 magic states can be detected by the X-basis measurements. If any X-basis measurement outcome is -1, all qubits are discarded and the distillation protocol is restarted. Suppose that every input magic state suffers Pauli Z error with probability p_i; then the error rate of the output magic state is $p_{\text{out}} \approx 35p_i^3$ since there are 35 combinations of three faulty magic states that cannot be detected. If the input magic states suffer X, Y, Z Pauli errors with probabilities p_x, p_y, p_z, respectively, the output error rate is $p_{\text{out}} \approx 35[p_z + \frac{(p_x+p_y)}{2}]^3$ [29].

Typically, the initial input magic states are produced by a non-fault-tolerant procedure called state injection. This procedure injects an arbitrary physical state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ into the logical state $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$. The state injection is crucial since the fidelity of the distilled magic states strongly depends on the quality of the initial magic states. For example, in the 15 to 1 distillation protocol, if the error rate of the initial states p_i is reduced to p_i/n, the error rate of the output state is reduced by around $n^{(3^k)}$ times for k rounds of distillation.

D. Qubit layout and circuit-level noise model

Here we introduce the basic assumptions about the qubit layout. The data qubit is placed in each vertex of the primal lattice of the triangular color code, and two syndrome qubits are in each colored face for the stabilizer measurements. One of the syndrome qubits is initialized to $|0\rangle$ for measuring the Z-type stabilizer and the other is initialized to $|+\rangle$ for measuring the X-type stabilizer. The syndrome qubits are coupled with data qubits in the corresponding faces by CNOT gates. Those assumptions run through the following discussions of the color code decoding, lattice surgery, and magic state injection.

It is known that error correcting performance of QEC codes largely depends on noise models. Throughout this paper, the noise model we considered is the circuit-level depolarizing noise model. The depolarizing error channels are defined as

$$\mathcal{E}_1(\rho_1) = (1-p)\rho_1 + (p/3)\sum_{P\in\{X,Y,Z\}} P\rho_1 P,$$

$$\mathcal{E}_2(\rho_2) = (1-p)\rho_2 + (p/15) \times \sum_{\substack{P_1,P_2\in\{I,X,Y,Z\},\\ P_1\otimes P_2\neq I\otimes I}} P_1\otimes P_2\rho_2 P_1\otimes P_2, \tag{1}$$

where ρ_1 and ρ_2 are single-qubit and two-qubit density matrices, respectively, and p is the physical error rate.

When simulating a noisy quantum circuit, we approximate every noisy operation by an ideal operation, followed by the depolarizing error channel $\mathcal{E}_1$ or $\mathcal{E}_2$. Specifically, the depolarizing error channels are added to the circuits by the following rules:

(1) add $\mathcal{E}_1$ after preparing each $|0\rangle$ or $|+\rangle$ state,
(2) add $\mathcal{E}_1$ after each idle operation,
(3) add $\mathcal{E}_1$ before each measurement in Z or X basis,
(4) add $\mathcal{E}_2$ after each CNOT gate.

A noisy quantum circuit for preparing a Bell state is shown in Fig. 5. Note that if a qubit in some step is not acted upon by any of the preparation, two-qubit gate, or measurement while operations of other qubits are being performed, it is assumed to be applied to an idle operation and suffer the error channel $\mathcal{E}_1$. In fact, errors from idle operations account for a large portion when we simulate a high-weight stabilizer measurement circuit of the color code.

III. IMPROVED COLOR CODE DECODING

This section describes the improved color code decoding strategy and presents numerical results. Our improved decoding strategy is based on the projection decoder [33,34]. The variation is that we execute the MWPM algorithm on the graphs with error-rate-related weights (referred to as decoding graphs below), rather than the original dual lattice $\mathcal{L}^* \times n$. In Sec. III A, we discuss how to construct the decoding graphs and the criterion that a proper CNOT schedule (i.e., the order of the CNOT gates in the QEC circuits) needs to meet. In Sec. III B, we explain the entire process of the decoding algorithm and give the numerical results.

A. Decoding graphs and CNOT schedules

When decoding topological QEC codes by the MWPM algorithm, the performance of the decoder is affected by the matching weights. For example, in the surface code, the MWPM algorithm is typically applied in a graph with weights where the weight of the edge is related to the probability of the given pair of syndrome changes [15,51–54]. Reference [51] has calculated the probabilities of the six types of edges by categorically analyzing the error in the surface code QEC circuit.

By introducing the error-rate-related weights, the decoding performance increases substantially. Therefore, this inspires us to consider introducing the error-rate-related weights in the color code decoding graph. However, constructing such decoding graphs of color codes faces two main challenges:

(a) The QEC circuit of the color code is deeper and the qubit array is more complex compared to that of surface codes, which makes analyzing the edge error rate categorically more difficult.

(b) For the surface code, by setting the proper CNOT schedule, a single error in the circuit always causes two syndrome changes or less. Here a single error means the Pauli error from one depolarizing error channel in the noisy circuit. However, for the color code, because of the high-weight stabilizer measurements and error propagations, a single error may cause syndrome changes of more than two, in which case an error no longer corresponds to just one edge in the decoding graph.

To address challenge (a), we designed an automated procedure to count the syndromes of various errors in the QEC circuit. First, we construct three 3D decoding graphs $\mathcal{G}_C$ ($C \in \{RG, RB, GB\}$), where the vertical dimension is the time layers, and in each time layer there is a 2D graph. We assume that the total number of time layers in the decoding graph $\mathcal{G}_C$ is $d+1$, which corresponds to d rounds of noisy QEC circuits and one round of perfect QEC circuit. The vertices of the decoding graph are the same as those in the 3D restricted lattice $\mathcal{L}_C^* \times (d+1)$, but the edges are different.

Second, the edges in the decoding graph $\mathcal{G}_C$ are determined by the single error in the QEC circuit. To establish the relation between a single error and an edge, we simulated a perfect circuit attached to a single error ϵ in the different positions in the QEC circuit. This single error could originate from channel $\mathcal{E}_1$ ($\epsilon \in \{X, Y, Z\}$) or from channel $\mathcal{E}_2$ ($\epsilon \in \{I, X, Y, Z\}^{\otimes 2}/I^{\otimes 2}$). For each possible position and type of single error, the procedure simulates the QEC circuit one time and records the syndrome changes in each simulation.

Suppose that in the decoding graph $\mathcal{G}_C$, these recorded vertices are connected in pairs by some rules (see Appendix A for details) and the error ϵ leads to the edge e. Then the weight of e is assigned $w_e = -\log \sum_\epsilon p_\epsilon(e)$, where $p_\epsilon(e) = p/3$ if ϵ is a single-qubit error, and otherwise $p_\epsilon(e) = p/15$.

Now let us consider challenge (b): how to reduce the number of syndrome changes caused by a single error. In fact, the problem can be dealt with by a proper CNOT schedule. Intuitively, we hope that the multiple errors resulting from a single error propagation are located in neighbor vertices on a face of the primary lattice, so that some syndrome changes can cancel each other out. For example, let us consider the measurement circuit of the stabilizer generator on a hexagonal face. If an X error in syndrome qubit propagates to two neighbor data qubits, the number of syndrome changes is two. In contrast, if two error data qubits are not neighbors, the number of syndrome changes is four. Based on this intuition, we propose a criterion that a proper CNOT schedule needs to meet.

Suppose l_i is the position label of the data qubit where the ith CNOT gate ($i = 1, 2, 3, \ldots$) is applied, with $l_i = 1, 2, \ldots, N$ representing integers fixed on the weight-N stabilizer that increase clockwise (see Fig. 6). A proper CNOT schedule is defined as follows: given any $i \in \{1, 2, \ldots, N\}$, $\forall a, b \in \{l_{i+1}, l_{i+2}, \ldots, l_N\}$, we have $|a-b| \bmod N \leqslant N-i$, where $x \bmod N$ is the modulo function ranged in $[0, N)$. In other words, the criterion guarantees that the syndrome qubit's error after the ith CNOT gate always propagates to the data qubits whose positions $\{l_{i+1}, l_{i+2}, \ldots, l_N\}$ are $N-i$ neighbor positions. It can be verified that when N takes values of 2, 4, 6, or 8 (corresponding to all the stabilizer generators involved in this paper), the syndrome changes of these $N-i$ errors on each 2D restricted lattice do not exceed 2.

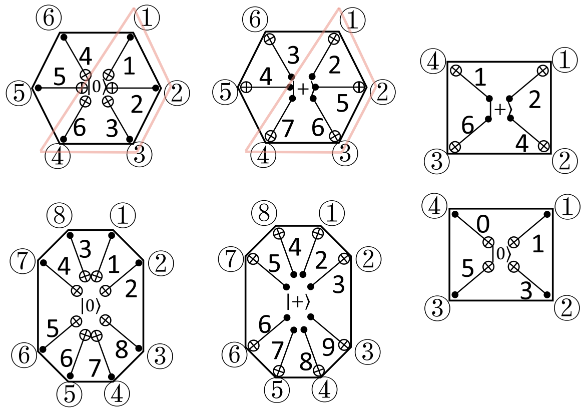


FIG. 6. CNOT schedules of stabilizers with different weights used in color code QEC circuits and lattice surgery. The outside and inside numbers indicate position labels and time orders, respectively. The CNOT schedules for weight-4 stabilizers on the triangular boundary can be derived by partitioning the CNOT schedules for weight-6 stabilizers. For example, the red trapezoid in the figure corresponds to the weight-4 stabilizer on the left boundary of the triangle. Similarly, the CNOT schedules of the stabilizers on the right and bottom boundaries of the triangle can be derived by partitioning the hexagon into trapezoids in different orientations.

Previous work [33] identified the optimal CNOT schedule with a specific physical error rate and qubit layout through numerical testing, which exactly meets the criterion. Therefore, we continue to use this CNOT schedule of weight-6 stabilizers in Fig. 6. It should be noted that the criterion we proposed also applies to higher-weight stabilizer measurements, such as the weight-8 stabilizers used in the lattice surgery (see Sec. IV and Fig. 6).

Once the CNOT schedule is given, we can explain how to connect the syndrome changes in pairs of a single error. There are five types of syndrome changes of a single error using the well-selected CNOT schedule. As explained in Appendix A and Fig. 7, the edges $e \in E(\mathcal{G}_C)$ in the three decoding graphs are determined by the circuit error that connects the syndrome change pairs. Except for one type in the bottom-left corner of Fig. 7, only two syndrome changes occur in each decoding graph, at most. Moreover, if the error propagates to the boundary data qubits, one of the vertices of the edge may come from the boundary vertex set $\{v_R^{(t)}, v_G^{(t)}, v_B^{(t)}\}$, where t is the time layer in which the error occurs.

B. Improved color code decoding algorithm

In this section, we show the whole decoding process of triangular color codes. Without loss of generality, we assume that one corrects X-type errors using the Z-type syndromes.

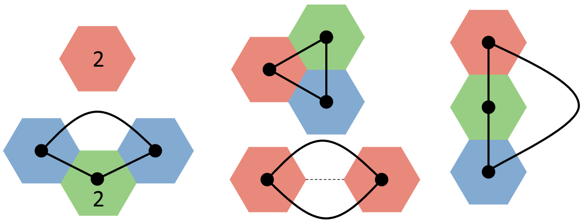


FIG. 7. Five possible syndrome changes (colored faces) and their connections (solid lines) of a single error in the color code QEC circuit with the proper CNOT schedule. We only show the projection of the faces, which means that the syndrome changes may come from different time layers. A face labeled 2 indicates that the syndrome changes twice in two consecutive time layers and corresponds to a vertical connection.

Roughly speaking, the decoding algorithm can be divided into three parts. First, we construct three decoding graphs and apply the MWPM algorithm on them. Then the matching results will be mapped to the 3D lattice $\mathcal{L}_C^* \times (d+1)$. Finally, the error data qubits are determined by combining the matching results.

The inputs of the decoding algorithm are code distance d, physical error rate p, and syndrome changes $\sigma \subseteq V[\mathcal{L}^* \times (d+1)]$. In order to get the error correction set $R \subseteq F(\mathcal{L}^*)$, the decoding algorithm is performed in the following steps:

(1) Input d, p and syndrome changes σ, and construct the 2D dual lattice $\mathcal{L}^*$. For $C \in \{RG, RB, GB\}$, construct the 3D dual lattice $\mathcal{L}^* \times (d+1)$, the 3D restricted lattice $\mathcal{L}_C^* \times (d+1)$, and the decoding graph $\mathcal{G}_C$.

(2) For $C \in \{RG, RB, GB\}$, apply the MWPM algorithm on $\mathcal{G}_C$ to pair up the vertices in $\sigma_C = \sigma \cap V(\mathcal{G}_C)$ and obtain a path set S_C.

(3) For all s in S_C, check whether the edge $e \in E[\mathcal{G}_C]$ satisfies $e \in E[\mathcal{L}_C^* \times (d+1)]$. If not, replace e on s with the shortest path connecting endpoints of e in $\mathcal{L}_C^* \times (d+1)$.

(4) Combine path sets $S = S_{RG} \cup S_{RB} \cup S_{GB}$. For s_1, s_2 in S, if s_1, s_2 can be concatenated, replace s_1, s_2 of S with their concatenation $s_1 + s_2$. Repeat this step until all the paths in S cannot be concatenated.

(5) For path $s \in S$, project s to the 2D lattice $\mathcal{L}^*$. The projection proj(s) will divide $\mathcal{L}^*$ into two face sets F_1 and $F[\mathcal{L}^*]\backslash F_1$, and select the smaller one as the correction set R_s. Output the total correction set $R = \oplus_s R_s$, where $\oplus$ is modulo 2 addition.

The algorithm essentially follows the steps outlined in Ref. [33]. The distinction is that we introduce decoding graphs with error-rate-related weights and map it back to a 3D lattice $\mathcal{L}_C^* \times (d+1)$ in the third step.

After the fourth step, the paths in S are either enclosed or have two endpoints from the boundary vertex set $\tilde{V} = \{v_x^t | v_x \in \{v_R, v_B, v_G\}, t = 1, 2, \dots, d+1\}$. Therefore, after the projection in the fifth step, the path always divides $\mathcal{L}^*$ into two complementary regions.

We also remark that the entire algorithm can be efficiently executed in polynomial time. First, when constructing the decoding graph $\mathcal{G}_C$, it is necessary to simulate approximately $O(m)$ QEC circuits, where m is the number of data or syndrome qubits. This simulation frequency is proportional to the number of noise channels in the QEC circuit and, consequently, proportional to the number of qubits. The time complexity for simulating one QEC circuit is $O(m^2)$ [55], resulting in an overall time complexity of $O(m^3)$. For given d and p, the decoding graphs only need to be generated one time and can be repeatedly used. Additionally, the primary time consumption in the decoding algorithm occurs in the MWPM algorithm, which has a time complexity of $O(m^3 \log m)$ [36]. Overall, the time complexity of the decoding algorithm is $O(m^3 \log m)$.

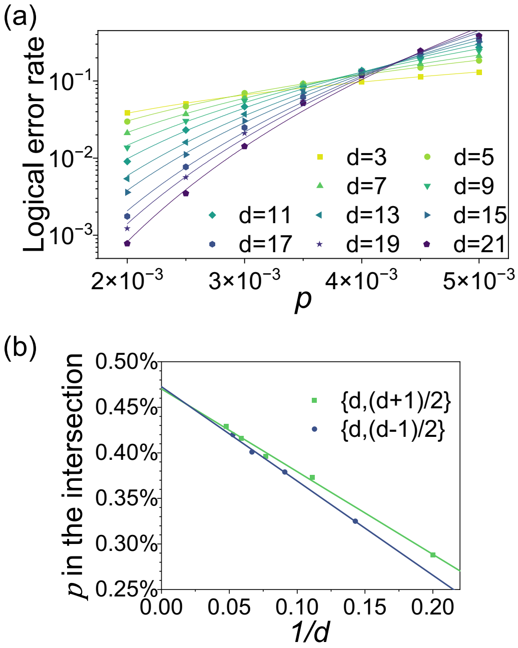


FIG. 8. (a) Logical error rates P_L of the triangular color code under the circuit-level noise model for various code distances d. The curves are fitted by the ansatz $P_L = \alpha p^\beta$, where α and β vary with d. (b) The intersections of pairs of curves of P_L with distances $\{d, (d+1)/2\}$ and $\{d, (d-1)/2\}$. From the linear extrapolation of the data, the threshold is around 0.47%.

We simulate the logical error rates P_L of the color code with code distance from $d = 3$ to $d = 21$ and fit the curves by ansatz $P_L = \alpha p^\beta$, where parameters α and β vary with different d [see Fig. 8(a)]. Theoretically, the threshold is the value of the physical error rate p of the intersection of the curves when $d \to \infty$. Here the threshold is progressively estimated by the intersections of two classes of the logical error rate curves, similar to the method in Ref. [33]. We select the curves with distances $\{d, (d+1)/2\}$ and $\{d, (d-1)/2\}$, since their slopes differ quite remarkably, to identify the intersections [see Fig. 8(b)]. From a linear extrapolation of the data, the threshold is around 0.47%. For the convenience of future research, we provide a formula for the average logical error rate per round of QEC circuit, $\bar{P}_L = 0.018(p/0.47\%)^{d/3+0.04}$, obtained through fitting data with $d \geqslant 11$.

Intuitively, our strategy improves upon the decoder presented in Ref. [33]. However, we emphasize that the threshold result here cannot be directly compared to the 0.37% threshold in Ref. [33] because our noise model is slightly weaker. Nonetheless, we provide some simulation results under the

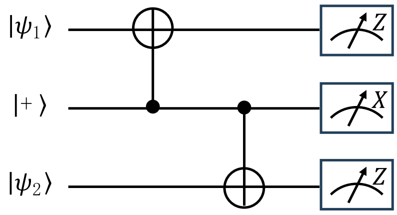


FIG. 9. Circuit to measure $X \otimes X$ and $Z \otimes Z$ using a $|+\rangle$ state as ancilla. The result of $X \otimes X$ is the measurement outcome on $|+\rangle$, and the result of $Z \otimes Z$ is the product of measurement outcomes on $|\psi_1\rangle$ and $|\psi_2\rangle$. After the measurements, all the states will be discarded. Note that this circuit can be viewed as the inverse of the circuit in Fig. 5.

same noise model in Appendix D to directly demonstrate the performance improvement.

IV. CIRCUIT-LEVEL DECODING OF COLOR CODE LATTICE SURGERY

Although, in principle, the lattice surgery of triangular color codes is feasible, it still requires a circuit-level decoding strategy for its practical application. In this section, we focus on the decoding process during the measurement of two-body Pauli operators using lattice surgery. Section IV A presents the detailed steps of the decoding algorithm. Section IV B provides numerical results and discusses the applicability of the algorithm.

A. Decoding algorithm

Let us clarify the decoding strategy in the specific example of Fig. 2(b), where two-qubit logical operators $X_L \otimes X_L$ and $Z_L \otimes Z_L$ are measured in parallel. Suppose the logical qubits are in the PBC model; then, d rounds of QEC circuits will be implemented alternately in the merged and split lattice. Here, we consider a section in this process with a total of $2d + 1$ layers. Namely, the first $(d + 1)/2$ layers and the last $(d + 1)/2$ layers are split lattices, while the middle d layers are merged lattices. In the QEC circuits, we continue to use CNOT schedules of weight-6 stabilizers, and give weight-8 and weight-4 CNOT schedules in Fig. 6, which meets the criterion proposed in Sec. III A.

At the beginning and end of the merged step, the Bell pairs are prepared and measured by the circuits in Figs. 5 and 9. Note that there are enough syndrome qubits with our qubit layout to act as the ancilla in these two circuits since the number of syndrome qubits is approximately $2n_f$ and the number of Bell states is approximately n_f, where n_f is the number of the faces in the middle region. Also, remarkably, at the beginning and end of the merged step, the syndromes of green and blue stabilizers are inferred by the product of stabilizers of the Bell states after initialization or measurements, making them noiseless. On the other hand, the red syndromes are random during these two steps, meaning they have no determined initial and final values. After measuring Bell state measurements, the weight-8 stabilizers will transform into weight-4 stabilizers in the next QEC round. Therefore, in the first QEC round of the split lattice, the syndromes of the green or blue stabilizers at the boundaries between regions that used to be weight-8 stabilizers in the merged lattice need to be multiplied by the outcomes of two Bell pair measurements.

Before introducing the decoding algorithm, we specify some notations about lattices and vertices. We use $\mathcal{L}^m$ ($\mathcal{L}^s$), $\mathcal{L}_C^m$ ($\mathcal{L}_C^s$), and $\mathcal{L}_C^{ls}$ to denote the 2D merged (split) dual lattice, the 2D restricted merged (split) lattice, and the 3D restricted lattice, respectively, where $C \in \{RG, RB, GB\}$. Note that all the lattices discussed in this section are dual lattices, so we omit the superscript asterisk in lattice labeling. Additionally, when referring to a 3D lattice, we assume that there are only vertically connected edges in the vertical direction, without diagonal edges (diagonal edges will appear in the decoding graphs). In the 3D restricted lattice $\mathcal{L}_C^{ls}$, the vertices of d layers in the middle are from the merged lattice $\mathcal{L}_C^m$ and the vertices in other layers are from the split lattice $\mathcal{L}_C^s$ (see Fig. 10). In particular, attention is paid to the red vertices in the lattice $\mathcal{L}^m$. The red vertices in region k ($k = 1, 2, 3$) form a vertex set denoted by V_R^k. Beyond the three regions, there are some weight-2, weight-4, or weight-8 stabilizers at the boundaries between regions, corresponding to vertices in set V_K^m ($K \in \{R, G, B\}$ indicating the color). We also define a boundary vertex set: $\tilde{V} = \{v_x^t | v_x \in \{v_B, v_G\} \cup V_R^m, t = 1, 2, \ldots, 2d + 1\} \cup \{v_x^t | v_x \in V_R^2, t = (d + 3)/2 \text{ or } (3d + 3)/2\}$.

There are two goals of the lattice surgery decoding algorithm. One is obtaining the correction set $R \subseteq F(\mathcal{L}^m)$ as accurately as possible and the other is obtaining the correct measurement results of $X_L \otimes X_L$ and $Z_L \otimes Z_L$. Overall, our decoding strategy has three main parts. At the beginning, we construct three decoding graphs $\mathcal{G}_C$ ($C \in \{RG, RB, GB\}$) with $2d + 1$ layers [see Fig. 10(b)]. The edges and their weights in $\mathcal{G}_C$ can be obtained by simulating lattice surgery circuits by the automated procedure using a similar method as in Sec. III B. Then we input the syndrome changes σ and apply the MWPM algorithm. Likewise, we map the matching results to 3D lattice $\mathcal{L}_C^{ls}$. Lastly, each path is projected on the 2D merged dual lattice $\mathcal{L}^m$.

Different from Sec. III B, after the projection, there are three types of paths instead of two. The first is enclosed and the second starts and ends with v_G or v_B. These two types of paths can divide the lattice into two complementary regions and we select the smaller one as the correction. The third type is the path that starts or ends with the vertices in $V_R^m \cup V_R^2$ and is not enclosed [see the green line in Fig. 10(a)]. This results from the indeterminate measurement results in V_R^m and V_R^2, as their initial values are random and there is no perfect measurement in the final round. The syndromes on $V_R^m \cup V_R^2$ correspond to $X \otimes X$ (or $Z \otimes Z$) errors on several Bell pairs (see Fig. 2). Therefore, the incorrect measurements on $V_R^m \cup V_R^2$ are equivalent to errors on several Bell pairs after decoding. However, as described in Appendix C, the impact of $X \otimes X$ (or $Z \otimes Z$) errors within Bell pairs on the decoding outcome is trivial. Hence, in the decoding, the lattice $\mathcal{L}^m$ is divided into several subareas by the blue lines and the outermost circle [see Fig. 10(a)]. Each subarea corresponds to a Bell pair or a logical qubit and the error suffusing any subareas is trivial or a logical error. We find the smaller regions separated by the path in each subarea as the correction [see green circles in Fig. 10(a)].

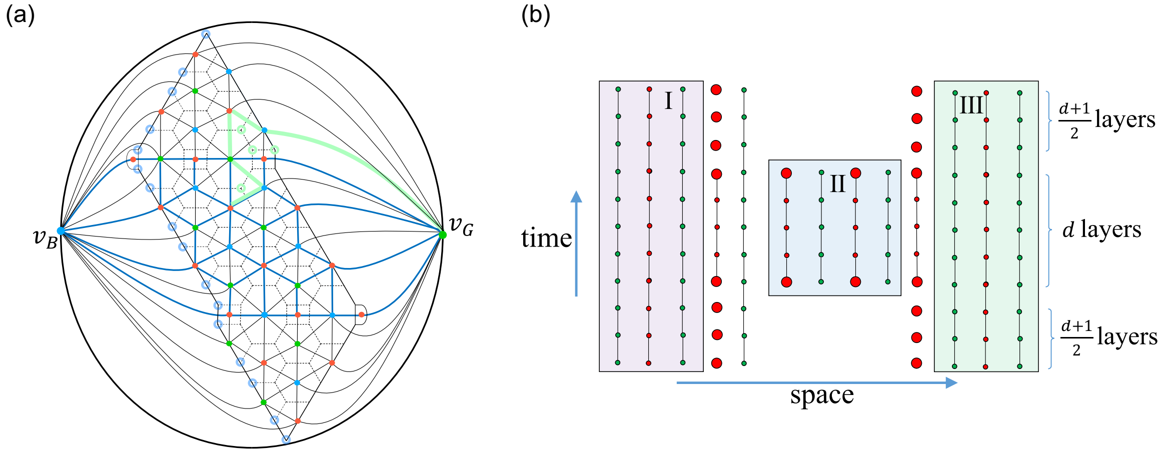


FIG. 10. (a) The merged dual lattice $\mathcal{L}^m$ of lattice surgery of color code lattice surgery for measuring $X_L \otimes X_L$ and $Z_L \otimes Z_L$ in parallel. The lattice is divided into several subareas by the blue lines and the outermost circle. An example of the path that ends inside the lattice and corresponding correction are illustrated by green lines and circles. Additionally, the blue circles indicate all the data qubits on the left boundary, which support the boundary operator b_B. (b) Vertices in decoding graph $\mathcal{G}_{RG}^{ls}$ (or lattice $\mathcal{L}_{RG}^{ls}$). Two dimensions in space are compressed into one. The boundary vertex v_G is not shown, although it is also in each layer of $\mathcal{G}_{RG}^{ls}$ or $\mathcal{L}_{RG}^{ls}$. In our simulation, regions I and III are split in the first and last $(d+1)/2$ layers and merged by region II in the middle d layers. The magnified vertices represent red vertices in the boundary vertex set $\tilde{V}$. Beyond the three regions, the vertices in $V_R^m \cup V_G^m$ are retained in each layer of the split steps. To avoid clutter, only vertical edges are shown, while horizontal edges as well as diagonal edges in $\mathcal{G}_{RG}^{ls}$ or $\mathcal{L}_{RG}^{ls}$ are omitted.

As mentioned, the measurement outcome of $X_L \otimes X_L$ (or $Z_L \otimes Z_L$) is also required to be determined. We use the product of all measurement outcomes of the starred stabilizers in the $(d+3)/2$ layer as the raw result. Apparently, the raw result is inaccurate unless there are no errors affecting the measurement results. Therefore, if any syndrome change of the starred stabilizer occurs before or in the $(d+3)/2$ layer, we flip the raw result. In addition, we specifically consider the third type of paths mentioned above. The syndrome of the correction R_s obtained from the third type of paths and the actual syndrome change may be different in some boundary vertices from $V_R^m \cup V_R^2$. If such vertices are before or in the $(d+3)/2$ layer and the number of these vertices is odd, the raw result also needs to be flipped.

In summary, the decoding algorithm is applied as follows:

(1) Input d, p and syndrome changes σ and construct the 2D merged dual lattice $\mathcal{L}^m$. For $C \in \{RG, RB, GB\}$, construct the 3D restricted lattices $\mathcal{L}_C^{ls}$ and decoding graph $\mathcal{G}_C^{ls}$ by the automated procedure in the similar way as before.

(2) For $C \in \{RG, RB, GB\}$, apply the MWPM algorithm on $\mathcal{G}_C^{ls}$ to pair up the vertices in $\sigma_C = \sigma \cap V(\mathcal{G}_C^{ls})$ and obtain a path set S_C.

(3) For path $s \in S_C$, check whether the edge $e \in E(\mathcal{G}_C^{ls})$ on s satisfies $e \in E(\mathcal{L}_C^{ls})$. If not, replace e with the shortest path connecting endpoints of e in $\mathcal{L}_C^{ls}$.

(4) Combine path sets $S = S_{RG} \cup S_{RB} \cup S_{GB}$. For s_1, s_2 in S, if s_1, s_2 can be concatenated, replace s_1, s_2 of S with their concatenation $s_1 + s_2$. Repeat this operation until all the paths in S cannot be concatenated.

(5) For path $s \in S$, project s to $\mathcal{L}^m$. If proj(s) divide $\mathcal{L}^m$ into two regions, select the smaller one as the correction. Otherwise, select the smaller region separated by proj(s) in each subarea as the correction R_s. Output the total correction set $R = \oplus_s R_s$.

(6) Initialize the measurement outcome of $X_L \otimes X_L$ (or $Z_L \otimes Z_L$) m_o as the product of measurement results of the starred stabilizers in the $(d+3)/2$ layer. For path $s \in S$, if an odd number of syndrome changes of starred stabilizers in s occurs before the $(d+3)/2+1$ layer, replace m_o with $-m_o$. For s that has an endpoint in $\{v_x^t | v_x \in V_R^m \cup V_R^2, t \leqslant (d+3)/2\}$, compare the syndrome of correction R_s in the lattice $\mathcal{L}^m$ and the projections of actual syndrome changes. If they differ in an odd number of vertices, replace m_o with $-m_o$. Output m_o as the measurement outcome.

B. Numerical results and further discussions

We use Monte Carlo simulation to test the performance of our algorithm. For the results in the simulations, we say a spacelike error occurs if R causes a logical error, and a timelike error occurs if m_o is an incorrect outcome and no spacelike errors occur. The numerical results of the error rates of these two type of errors are shown in Fig. 11. The error rate curves are also fitted by the ansatz αp^β. In principle, we estimate that the curves' spacelike error rates have the parameters $\beta \sim d/3$ since our decoding algorithm is based on the projection decoder, which admits that it can correct all the errors of weight, at most, $d/3$ (up to an additive constant) [33]. On the other hand, the timelike error curves are expected to have the parameters $\beta \sim d/2$ since they mainly result from the inaccurate measurement results of d rounds of QEC circuits. Therefore, in Fig. 11(e), we use the lines with fixed slopes to fit the parameters β of the curves, which indicates that the spacelike effective code distance is less than that of the timelike one. We also find that the timelike error rate is generally one to two orders of magnitude smaller than the spacelike error rate, implying that the dominated logical error in color code lattice surgery is the spacelike type. Note

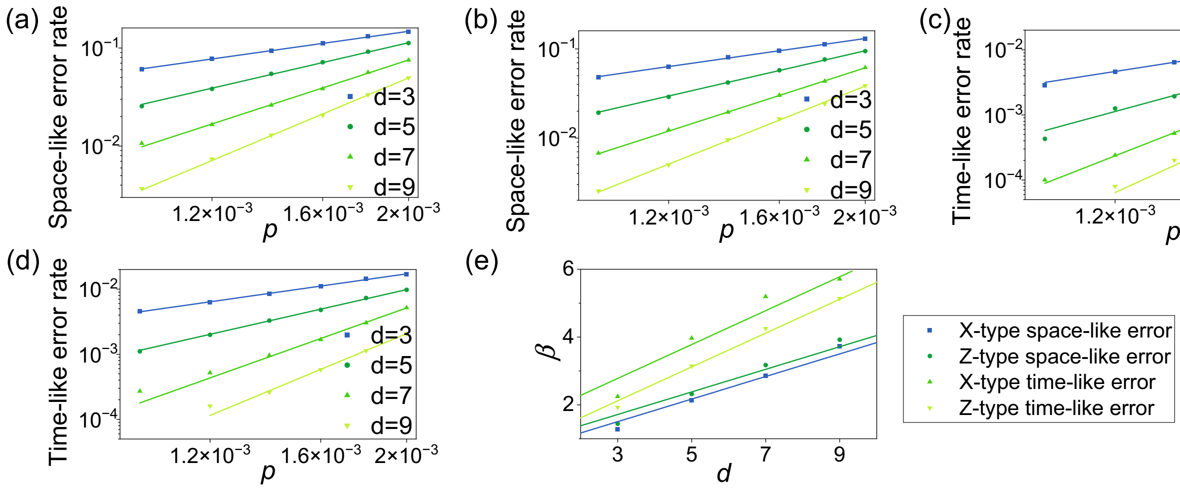


FIG. 11. Space-like error rates and time-like error rates in the lattice surgery for various code distances d. All the curves of error rates are fitted by the ansatz αp^{β}. (a), (b) X-type and Z-type space-like error rates. (c), (d) X-type and Z-type time-like error rates. (e) Parameter β of the curves in (a-d) for various code distances d. The parameters β in space-like error rates and time-like error rates are fitted by $\beta = d/3 + c_1$ and $\beta = d/2 + c_2$ respectively.

that the rates of X-type and Z-type errors are not equal in our simulation, and we speculate that this is due to the order of the X-type and Z-type stabilizer measurements.

It should be emphasized that since our algorithm is based on the projection decoder [33,34], it may not be applicable to arbitrary forms of lattice surgery on color codes. Therefore, in the following, we will use the symmetry theory from Ref. [56] to explain why our decoding algorithm can decode lattice surgery in the previous section and its applicable scope.

For a subset Σ of the stabilizer group, we say it is a symmetry if $\prod_{S\in\Sigma} S_f = \mathbb{1}$. It is easy to verify that any error will cause an even number of syndrome changes within Σ, thus allowing the matching algorithm to be applied to the restricted lattice corresponding to Σ. Now consider the set of stabilizers shown in Fig. 10(a). Define the tensor product of X or Z operators on all data qubits on the left (or right) boundary as the boundary operator b_B (b_G). It is easy to verify the following relationships:

$$\prod_{f\text{ is not green}} S_f \times b_B = \mathbb{1},$$

$$\prod_{f\text{ is not blue}} S_f \times b_G = \mathbb{1},$$

and

$$\prod_{f\text{ is not red}} S_f \times b_G b_B = \mathbb{1}, \tag{2}$$

where f is the face in the primal lattice, and S_f is an X-type or Z-type stabilizer generator on the face f. Correspondingly, we can define three subsets of stabilizers with symmetry:

$$\Sigma_{RB} = \{S_f \mid f\text{ is not green}\} \cup \{b_B\},$$
$$\Sigma_{RG} = \{S_f \mid f\text{ is not blue}\} \cup \{b_G\},$$

and

$$\Sigma_{GB} = \{S_f \mid f\text{ is not red}\} \cup \{b_G, b_B\}. \tag{3}$$

It is clear that b_B and b_G correspond to the boundary vertices v_B and v_G in Fig. 10(a), and these three subsets correspond to the three restricted lattices in the decoding algorithm. This explains why a matching algorithm-based decoder can effectively handle the lattice surgery example discussed in the previous section.

In general, more examples of lattice surgery are presented in Refs. [19,38], which involve different types of boundaries between two regions. Reference [38] classifies these boundaries into color boundaries and Pauli boundaries. For lattices with only color boundaries, finding the above symmetries is straightforward (they correspond to three restricted lattices). Therefore, decoding algorithms based on the MWPM (including our decoding algorithm) can effectively handle the decoding problems of this type of lattice surgery. However, for lattices with Pauli boundaries or mixed colored and Pauli boundaries, it is unclear whether such symmetries exist. The study of decoding algorithms on such lattices remains an open problem.

V. STATE INJECTION

In this section, we propose the state injection protocol of the triangular color code. First, we describe the process and the principles of the protocol in Sec. V A. We also prove that the logical error rate of our protocol is lowest compared to the state injection protocol among all CSS codes. In Sec. V B, the performances of diverse postselection schemes are shown by numerical results.

A. State injection protocol

State injection is the process of obtaining an arbitrary logical state $|\psi_L\rangle$ from a physical state $|\psi\rangle$. Let us describe our state injection protocol of triangular color code. The protocol is executed in two steps. As illustrated in Fig. 12(a), in the first step, the top qubit is initialized to $|\psi\rangle$ and the other qubits are initialized to Bell states $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ in pairs. In

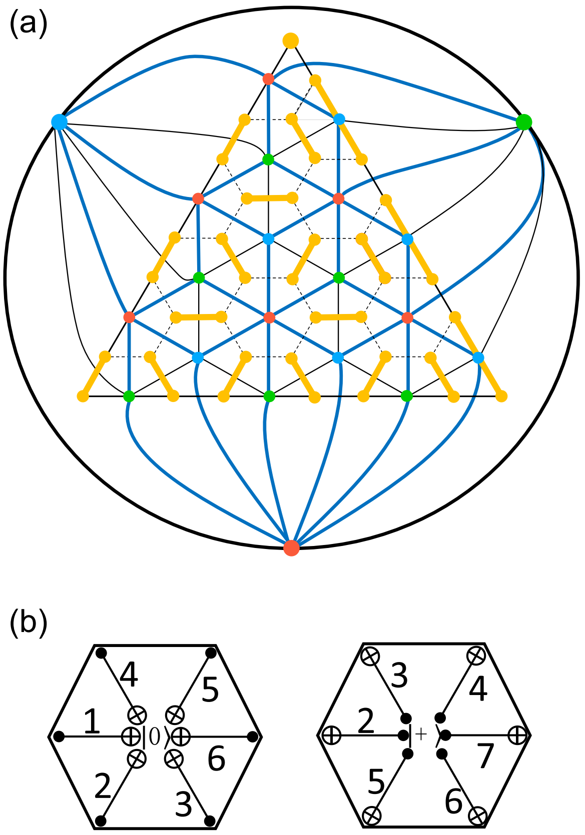


FIG. 12. (a) Prime (dashed lines) and dual (solid lines) lattice of state injection protocol. The qubits are prepared in Bell pairs (yellow), except for the injected qubit on the top. The lattice is divided into several subareas by the blue lines and the outermost circle. (b) The CNOT schedule in the state injection protocol that offers the lowest logical error rate.

the second step, the QEC cycles are performed with the CNOT schedule we give in Fig. 12(b), which is carefully selected from 12 different schedules (see Appendix B for details).

To prove that the final state is $|\psi_L\rangle$ after all noiseless operations, we can simply verify the following equation:

$$\begin{aligned} &|\psi_L\rangle\langle\psi_L|\prod_i \frac{I+S_i}{2}|\psi\rangle|\phi\rangle^{\otimes m} \\ &= \frac{I+\alpha X_L+\beta Y_L+\gamma Z_L}{2}\prod_i \frac{I+S_i}{2}|\psi\rangle|\phi\rangle^{\otimes m} \\ &= \left(\prod_i \frac{I+S_i}{2}\right)\frac{I+\alpha X_L+\beta Y_L+\gamma Z_L}{2}|\psi\rangle|\phi\rangle^{\otimes m} \\ &= \prod_i \frac{I+S_i}{2}|\psi\rangle|\phi\rangle^{\otimes m}, \end{aligned} \tag{4}$$

where we use $|\psi\rangle\langle\psi| = (I+\alpha X+\beta Y+\gamma Z)/2$, $|\psi_L\rangle\langle\psi_L| = (I_L+\alpha X_L+\beta Y_L+\gamma Z_L)/2$, and S_i are the stabilizer generators of the triangular color code. Hence, the final state $\prod_i (I+S_i)/2|\psi\rangle|\phi\rangle^{\otimes m}$ is the +1 eigenstate of $|\psi_L\rangle\langle\psi_L|$, i.e., $\prod_i (I+S_i)/2|\psi\rangle|\phi\rangle^{\otimes m} = |\psi_L\rangle$.

The state injection protocol is non-fault-tolerant because a single error in the circuit may cause a logical error that cannot be corrected by the decoding algorithm. More seriously, some errors cannot even be detected, which means postselection is also powerless against these errors. For example, the error occurring in the top qubit $|\psi\rangle$ before the first QEC cycle cannot be identified since the injection protocol is applicable to an arbitrary state, even though it is a faulty state.

Through exhaustive search, we find that in our protocol with postselections, there are two types of errors that cannot be detected. The first is the single-qubit error when preparing the physical state $|\psi\rangle$. The second is the two-qubit errors after some CNOT gates in the first QEC cycle, including $X\otimes I$, $Z\otimes Z$, and $Y\otimes Z$ errors after the CNOT gate between $|\psi\rangle$ and syndrome qubit $|+\rangle$, and $X\otimes X$ error after the CNOT gate between $|\psi\rangle$ and syndrome qubit $|0\rangle$. We will discuss in detail later why these errors are undetectable.

Therefore, with the postselection, the logical error rates of our state injection protocol are

$$P_L = \tfrac{4}{15}p_2 + p_I + \tfrac{2}{3}p_1 + O(p^2), \tag{5}$$

where p_1, p_2, and p_I are the error rate of single-qubit gates, CNOT gates, and initializing $|0\rangle$ or $|+\rangle$ state, respectively, which is consistent with Refs. [30,31]. We assume that $|\psi\rangle$ initialization is performed one time step before the first CNOT gate on $|\psi\rangle$ is applied. Hence there is no need to consider the contribution of idling errors of $|\psi\rangle$ to P_L.

As mentioned, the error rate of the input state of magic state distillation will remarkably affect the output error rates. In a rough version of the injection protocol in Ref. [33], the logical error rate after state injection is $6.07p + O(p^2)$ and $4p$ is from single-qubit errors. If the error rate of a two-qubit gate is assumed to be 10 times as the rate of a single-qubit error and we let $p_2 = p$, the parameter P_L in their protocol can be estimated as $P_L = (2.07 + 0.4)p + O(p^2)$ and P_L in our protocol is $4p/15 + p/6 + O(p^2)$. Taking the 15 to 1 distillation protocol as an example, we use $p_o = 35{P_L}^3$ to estimate the error rate of the distilled magic state. The output error rate p_o from our state injection is over two orders of magnitude lower than that of the rough injection protocol. It is important to emphasize that in the comparison, we neglected the higher-order terms of P_L. This was done for two reasons: first, only the first-order term of P_L is provided in Ref. [33], and second, our numerical results indicate that the contribution of higher-order terms to P_L is negligible for $p = 10^{-4}$ and below.

Compared to previous works in surface codes, our protocol admits a lower logical error rate (see the second column in Table I). Further, we claim that the logical error rate of our protocol is lowest for state injection protocols among all CSS codes. Let us clarify it clearly with the following theorem.

Theorem 1. For any CSS code with standard stabilizer measurement circuits, the logical error rate of a state injection protocol is at least $4/15p_2 + p_I + 2/3p_1 + O(p^2)$ under the circuit-level depolarizing noise model.

Here, standard stabilizer measurement circuits are the circuits using CNOT gates, measurement in Z (or X) basis, and $|0\rangle$ (or $|+\rangle$) as the ancilla for Z-type (or X-type) stabilizer measurements. Now let us prove the theorem by stating that several errors must not be detected in the state injection circuits. Without loss of generality, we suppose that $|\psi\rangle$ first couples with syndrome qubit $|+\rangle$ and then couples with syndrome qubit $|0\rangle$ in a QEC cycle. An obvious fact is that

TABLE I. Comparison of several state injection protocols.

Injection protocol	Logical error rate	Success rate of postselection	Other features
On planar surface codes [30]	$\frac{2}{5}p_2 + 2p_I + \frac{2}{3}p_1 + O(p^2)$	60% with $p = 10^{-3}$	Two steps where physical state is first injected to a $d = 7$ logical qubit
On rotated surface codes from corner [31]	$\frac{3}{5}p_2 + 2p_I + \frac{2}{3}p_1 + O(p^2)$	No data	Only gives results of $d \leqslant 5$
On rotated surface codes from middle [31]	$\frac{3}{5}p_2 + p_I + \frac{2}{3}p_1 + O(p^2)$	No data	Only gives results of $d \leqslant 5$
On triangular color codes (this work)	$\frac{4}{15}p_2 + p_I + \frac{2}{3}p_1 + O(p^2)$	Around 98% with $p = 10^{-4}$ and 88% with $p = 10^{-3}$	One step to the logical qubit with any large code distance

any error of $|\psi\rangle$ before the first stabilizer measurement is undetectable since $|\psi\rangle$ is arbitrary for an injection protocol. Therefore, the error of $|\psi\rangle$ initialization cannot be detected with rate $p_I + 2/3p_1 + O(p^2)$. In addition, the X error of $|\psi\rangle$ before the first CNOT gate between $|\psi\rangle$ and $|0\rangle$ but after the first CNOT between $|+\rangle$ and $|\psi\rangle$ is undetectable with the error rate of at least $p_2/15 + O(p^2)$, in which case the error is exactly $I \otimes X$ error after the first CNOT gate between $|\psi\rangle$ and $|+\rangle$. Further, utilizing the commuting relationship, a $Z \otimes Z$ error after the first CNOT gate between $|+\rangle$ and $|\psi\rangle$ is equivalent to an $I \otimes Z$ error before the stabilizer measurement, which is undetectable. Likewise, an $X \otimes X$ error after the first CNOT gate between $|\psi\rangle$ and $|0\rangle$ is also undetectable. Lastly, since $I \otimes X$ and $Z \otimes Z$ are undetectable, their product $Z \otimes Y$ after the first CNOT gate is also undetectable. In total, the logical error rate in a state injection protocol is at least $4/15p_2 + p_I + 2/3p_1 + O(p^2)$, which is exactly the result in our protocol. Hence, we have demonstrated that the logical error rate in our protocol serves as a lower bound for the logical error rate achievable by any state injection protocol among CSS codes.

B. Postselection schemes and numerical results

In the previous section, we theoretically analyze the leading-order term of the logical error rate in our protocol. In this section, we show the details of the postselection schemes and estimate the logical error rates in diverse schemes by numerical simulations. In our simulations, the noise model in Sec. II D continues to be used in the QEC circuits and the Bell state initialization circuits. We also assume that the initialization of $|\psi\rangle$ is perfect since errors in this step always cause logical errors. The logic error rates are counted after $d + 1$ rounds of QEC circuits.

Although the state injection protocol is non-fault-tolerant, it still requires decoding to suppress errors. We mainly follow the decoding strategy in Sec. III B, but make three minor changes. First, in the first step when constructing the decoding graphs, we take the errors in the Bell state initialization circuits into account. Second, the definition of boundary vertex set $\tilde{V}$ is modified as $\{v_x^t | v_x \in \{v_R, v_B, v_G\}, t = 1, 2, \dots, d + 1 \text{ or } v_x \text{ is red and } t = 1\}$. This means that the red vertices in the first time layer are boundary vertices because the outcomes of the stabilizer measurements in these positions are undetermined. Third, in the fifth step, the path s may start or end with the red vertex in the first time layer, whose projection cannot divide $\mathcal{L}^*$ into two parts. In this case, we use the method similar to that in Sec. IV A to divide $\mathcal{L}^*$ into several subareas to address this problem [see Fig. 12(a)].

In order to further suppress the logical error rate P_L, postselection schemes are introduced. The postselection means that if some syndromes are not as expected, we discard current states and restart the state injection protocol. Our results show that the postselections only need to be applied to five syndromes to minimize the leading-order term of P_L. The postselection scheme applied to these five syndromes is defined as Scheme A (see Fig. 13). Through exhaustive search, we find that there are 90 errors (excluding single-qubit errors in preparing the $|\psi\rangle$ state) that will lead to a logical error with the probability of $12.4p + O(p^2)$, and 86 of them can be detected by those five syndromes. The numerical results show that the logical error rate of postselection scheme A is close to $4/15p$ when $p = 10^{-4}$ [see Fig. 13(a)].

When $p = 10^{-3}$, however, the logical error rate is more than a dozen times of $4/15p$, since higher-order terms contribute a lot in this case. Therefore, it is necessary to apply postselection to more syndromes to suppress higher-order errors. To understand which syndromes are related to logical errors, we test the correlation coefficients of syndrome changes and the occurrence of logical errors by Monte Carlo simulations in a $d = 7$ triangular color code with $p = 10^{-3}$ [see Fig. 13(d)]. The correlation coefficient of events A and B is computed by

$$\rho_{AB} = \frac{|p(AB) - p(A)p(B)|}{\sqrt{p(A)[1 - p(A)]p(B)[1 - p(B)]}}, \tag{6}$$

where $p(A)$, $p(B)$, $p(AB)$ are the probabilities of A, B, and both A and B occurring, respectively.

According to these correlation coefficients, one can design the postselection scheme flexibly with the syndromes whose correlation coefficients are larger. Apparently, the postselection with more syndromes will lead to a lower success rate of postselection. Here we present an example with 35 postselected syndromes (scheme B) where the logical error rate is reduced by four times when $p = 10^{-3}$ compared to scheme A, while the success rate of postselection is still over 56%. More examples of postselection schemes with lower success rates are shown in Appendix B.

Contrary to previous works on state injections in surface codes, the success rate of postselections in our protocol does not sharply decay as the code distance increases, but is only

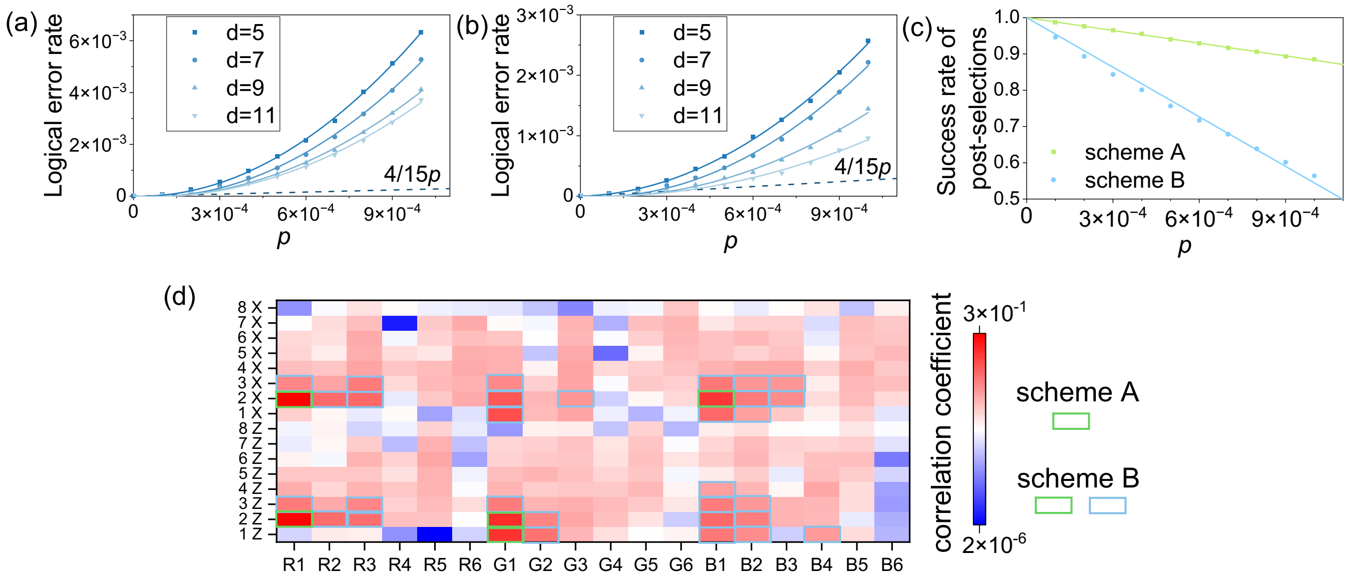


FIG. 13. (a),(b) The logical error rates of the state injection protocol in schemes A and B, respectively, where the preparation of the injected state is assumed to be noiseless since it always causes a logical error. (c) Success rates of the postselection as the linear function of the physical error rate p. The data are obtained from the state injection with $d = 11$, and the results with other code distances are very close to them. (d) Correlation coefficients between syndrome changes and the occurrence of logical errors. The label, say R2 and 3Z, means the syndrome of the second red Z-type stabilizer [in the order from top to bottom and left to right in Fig. 12(a)] in the third QEC round. The postselected syndromes in schemes A and B are framed, respectively.

linearly related to the physical error rate [see Fig. 13(c) and Table I]. On account of the fact that errors farther from the top qubit $|\psi\rangle$ are on a longer logical string (i.e., the data qubit string that connects left and right boundaries), the stabilizers of the postselected syndromes are located around the top qubit. Therefore, the number of postselected syndromes does not increase with the code distance, which means one can inject a physical state into a logical state with arbitrary code distance in one step, rather than two steps as in Ref. [30]. In addition, this feature provides our protocol with a high success rate of postselection, around 98% (88%) in scheme A when $p = 10^{-4}$ (10^{-3}).

VI. CONCLUSION AND OUTLOOK

In this paper, we solved several key problems on the road towards practical fault-tolerant quantum computation with 2D color codes, including improving the threshold of the triangular color code, decoding color code lattice surgery with color boundaries under circuit-level noise, and proposing an optimal state injection protocol. We believe these works are expected to promote the development of quantum computing based on color codes.

Our results show that more accurate matching weights can effectively improve the threshold of the color code. This improvement can also be applied to other matching-based decoders, for example, a concatenated MWPM decoder [57] whose scaling of the logical error rate is better than the projection decoder. Moreover, a better weight setting is to calculate the posterior probability of errors in the circuits by the belief propagation algorithm, known as the BP-matching algorithm, whose improvement has been demonstrated in surface codes [53,54]. In addition, the logical error rates of the color code lattice surgery in this work are higher than the results of the mature surface coding protocols [23,25]. Nevertheless, we believe that the results can be further improved by optimizing the circuits in the lattice surgery protocol, as in Ref. [37], narrowing the performance gap between color codes and surface codes in storage experiments through circuit optimization.

There are still many other challenges in quantum computing based on color codes that are left for future work. For example, the PBC model is probably not the final form of quantum computation based on color codes since it does not utilize the advantages of transversal Clifford gates on color codes. According to the specific quantum algorithm, a proper way to use color codes for quantum computing still needs to be developed.

In addition, we also note the color codes with other boundaries, such as square color codes or thin color codes [19,38], which are also candidates for color code quantum computing. These color codes encode more than one logical qubit in an individual patch and may further reduce the overhead under the special structured noise (e.g., biased noise). When the boundaries are of the colored type, we do not perceive fundamental difficulties in applying our lattice surgery decoding strategies to them. However, if Pauli boundaries are considered, decoding lattice surgery on such boundaries remains an unresolved issue. In particular, thin color codes with $d_z > d_x$ can reduce the cost of magic state distillation since the distillation error is more sensitive to the Z-type error [29]. Based on the feature of parallel measurements and our lattice surgery decoding algorithm, estimating and optimizing the time and space overhead of magic state distillation in a color-code-based quantum computation will also be an interesting question.

Lastly, our state injection protocol can enhance recent results on magic state preparation [49], further improving the fidelity of magic states. A possible future direction is the quantum computation with surface and color codes combined, where surface codes are used for computation and color codes are used for distillation.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 12034018) and Innovation Program for Quantum Science and Technology (Grant No. 2021ZD0302300).

APPENDIX A: AUTOMATED PROCEDURE FOR CONSTRUCTING DECODING GRAPHS

The automated procedure extracts error information of noisy circuits based on the circuit-level depolarizing noise model. In a noisy quantum circuit, we approximate every noisy operation by an ideal operation followed by the depolarizing error channel $\mathcal{E}_1$ or $\mathcal{E}_2$. We simulate the circuits in the Heisenberg representation [58], randomly inserting Pauli errors with corresponding probability. Since the circuits are Clifford circuits, according to the Gottesman-Knill theorem, they can be efficiently simulated in polynomial time [5,55].

As mentioned in the main text, in each time, the automated procedure simulates an ideal QEC circuit attached to a single error in the different channels in the circuit. If the error is located in channel $\mathcal{E}_1$, the error is X, Y, or Z. If the error is located in channel $\mathcal{E}_2$, the error is one of $P_1 \otimes P_2$, where $P_1, P_2 \in I, X, Y, Z$ and $P_1 \otimes P_2 \neq I \otimes I$.

In fact, there is no need to simulate every round of the QEC cycle since the effect of a single error is limited to two, the difference between two layers. For each position and type of error, we only simulate the circuit in two QEC rounds, where the error is only added in the first round and the circuit in the second round is perfect. We record the vertex sets corresponding to the syndrome changes in each simulation.

In the automated procedure, we connect the syndrome changes in pairs to form edges by the following rule. First, we connect the vertices that are projected into the same position, if they exist. Then we connect the remaining vertices with the same color, if they exist. Using the CNOT schedules in Fig. 6, there are, at most, two vertices remaining unmatched in each decoding graph after this step. If there are two vertices left, we connect them, and if there is only one vertex left, we check the syndrome of the error and connect the vertex with the proper boundary vertex.

Based on these connections, the edges in the first two layers of $\mathcal{G}_C$ are determined, denoted by $e_0 = (v_1^{t_1}, v_2^{t_2})$. All edges e_0 obtained from the simulations in two QEC rounds form an edge set β_0. Then one can construct the edge set β in $\mathcal{G}_C$ by β_0:

$$\beta = \left\{e = \left(v_1^{t_1}, v_2^{t_2}\right) \middle| v_1^{t_1}, v_2^{t_2} \in \mathcal{G}_C, e//e_0, e_0 \in \beta_0\right\}, \tag{A1}$$

where we say two edges $e_1 = (v_1^{t_1}, v_2^{t_2})$, $e_2 = (v_3^{t_3}, v_4^{t_4})$ are parallel in a 3D lattice if they have the same vertical projection and $t_3 - t_1 = t_4 - t_2$, notated by $e_1//e_2$.

Suppose edge $e_0 \in \beta_0$ is obtained from the syndrome changes of the single error ϵ; the error rate corresponding to e_0 is $p_\epsilon(e_0)$, where $p_\epsilon(e_0) = p/3$ if ϵ is the result of $\mathcal{E}_1$, otherwise $p_\epsilon(e_0) = p/15$. By accumulating $p_\epsilon(e_0)$ with different ϵ, we obtained the error-rate-related weight of the edge $e = (v_1^{t_1}, v_2^{t_2}) \in \beta$,

TABLE II. Logical error rates in different CNOT schedules with and without postselection.

CNOT schedule	P_L with postselection (to the leading order)	P_L without postselection (to the leading order)
0^+	$0.6p$	$10.6p$
0^-	$1.2\dot{6}p$	$13.2\dot{6}p$
1^+	$0.\dot{3}p$	$9.8\dot{6}p$
1^-	$0.5\dot{3}p$	$11.1\dot{3}p$
2^+	$0.\dot{3}p$	$10.8p$
2^-	$0.2\dot{6}p$	$12.4p$
3^+	$0.6p$	$11.5\dot{3}p$
3^-	$1.2\dot{6}p$	$12.2p$
4^+	$1.2p$	$12.0\dot{6}p$
4^-	p	$12.\dot{6}p$
5^+	$0.9\dot{3}p$	$13.6p$
5^-	p	$11.7\dot{3}p$

$$w_e = \begin{cases} -\log \displaystyle\sum_{\substack{e_0 \in \beta_0^{(1)}, \\ \text{and } e_0//e}} \sum_{\epsilon} p_\epsilon(e_0) & \text{if} \quad t_1 = t_2 = 1 \\ -\log \displaystyle\sum_{\substack{e_0 \in \beta_0^{(2)}, \\ \text{and } e_0//e}} \sum_{\epsilon} p_\epsilon(e_0) & \text{if} \quad t_1 = t_2 = d+1 \\ -\log \displaystyle\sum_{\substack{e_0 \in \beta_0, \\ \text{and } e_0//e}} \sum_{\epsilon} p_\epsilon(e_0) & \quad \text{otherwise,} \end{cases} \tag{A2}$$

where $\beta_0^{(1)} = \{e_0 | e_0 = (v_i^1, v_j^1) \in \beta_0\}$ and $\beta_0^{(2)} = \{e_0 | e_0 = (v_i^2, v_j^2) \in \beta_0\}$. Note that the edges in the time boundaries (i.e., the first and last layers) are considered individually in the first two cases.

APPENDIX B: DIFFERENT CNOT SCHEDULES AND MORE POSTSELECTION SCHEMES IN THE STATE INJECTION PROTOCOL

In order to find the optimal CNOT schedule of state injection, we test 12 kinds of CNOT schedules, as shown in Fig. 14. Note that 0^+ is the CNOT schedule used in color code QEC circuits and other schedules are obtained by flipping or rotating 0^+. We use these CNOT schedules since the flip or rotation operation will hold the relationship in the criterion of a proper CNOT schedule. The complete data of logical error rates to the leading order are shown in Table II.

In Table III, we tested more postselection schemes for the case of $p = 10^{-3}$ and $d = 11$. Recall that we prioritize syndromes with higher correlation coefficients for posterior selection. As more syndromes are postselected, P_L decreases to below p, with the cost of reduction in the success rate of postselection. Note that the logical error rate tends to stabilize with more syndromes being postselected. We infer that this

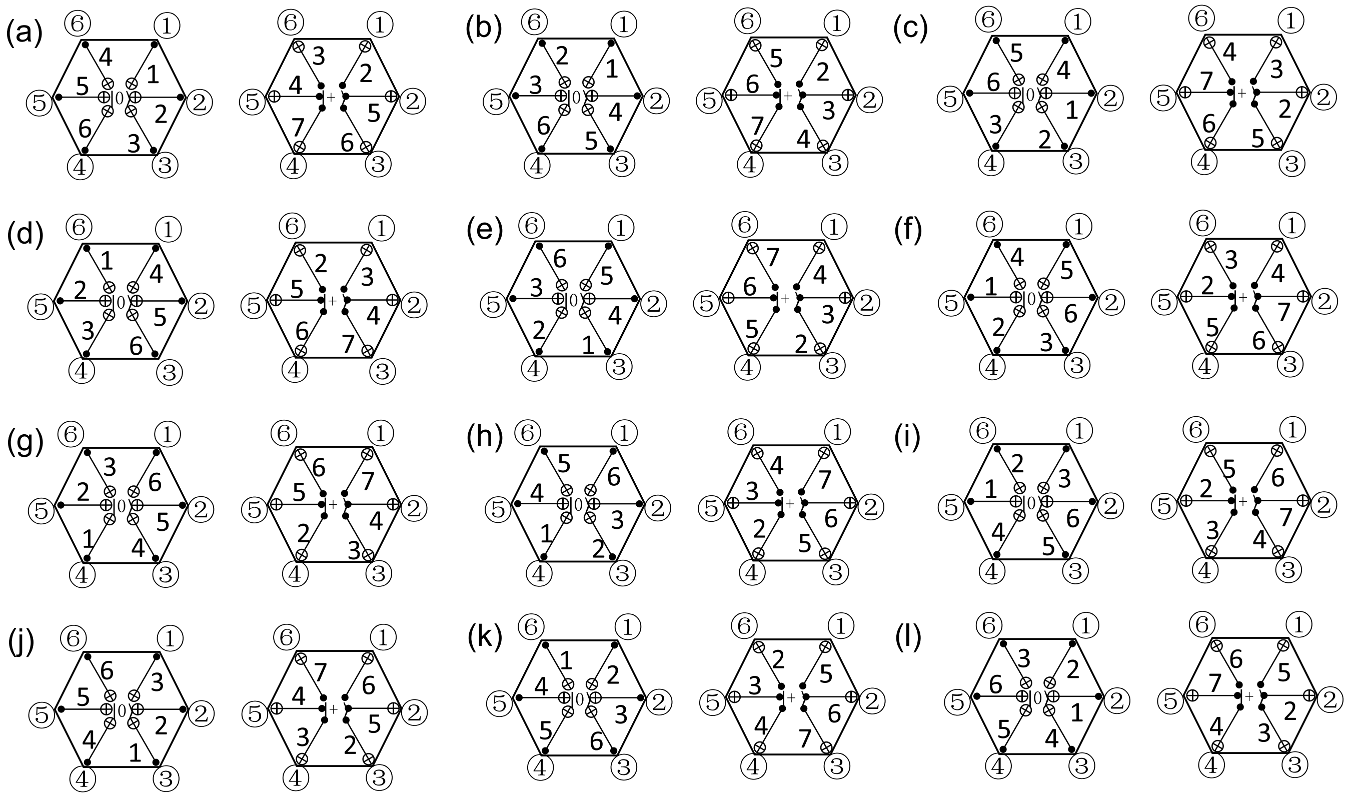


FIG. 14. Details of the 12 types of CNOT schedules that we considered in the state injection. (a)–(l) correspond to $0^+, 0^-, 1^+,..., 5^-$, respectively. The CNOT schedules with $+$ are obtained by rotating 0^+, and CNOT schedule k^- is the flip of k^+.

is due to the inherent logical error rate of $d = 11$ logical qubits, approximately 5×10^{-4}. Although postselection can reduce this value, the effect is not significant because the range of stabilizers we postselect is limited. Therefore, as the code distance of logical qubits continues to increase, achieving lower logical error rates than those listed in Table III is possible.

APPENDIX C: OTHER DETAILS OF THE SIMULATIONS

We use Monte Carlo simulation to estimate the logical error rates in different cases. In each time of simulation, QEC circuits are executed for $d + 1$ rounds in the color code decoding and state injection, and $2d + 1$ rounds (including the merged step and split step) in the lattice surgery. In all three situations, the circuit in the last round is assumed to be noiseless.

TABLE III. Logical error rates with other postselection schemes.

No. postselected syndromes	Logical error rate P_L	Success rate of the postselection
50	8.9×10^{-4}	40%
62	8.2×10^{-4}	34%
76	7.7×10^{-4}	27%
93	6.0×10^{-4}	21%
124	5.7×10^{-4}	14%
136	5.4×10^{-4}	12%
148	5.2×10^{-4}	10%
164	5.2×10^{-4}	9%

Another key question is how to determine if there is a logical error after decoding. In the color code decoding, we check whether the product of correction, R, and actual error ϵ is a logical operator (up to a stabilizer) by commutation relations.

In the lattice surgery, spacelike and timelike errors should be considered separately. Note that the spacelike logical error must anticommute with an odd number of starred stabilizers. The output correction R may cause a logical error as well as the stabilizers of several Bell states in the middle region since some boundary vertices are located inside the lattice. These stabilizers are trivial for our decoding because they always anticommute with an even number of starred stabilizers. The spacelike errors in the top and bottom logical qubits are equivalent since $X_L \otimes X_L$ and $Z_L \otimes Z_L$ have been measured. For the timelike error, we set all the red boundary vertices with the initial syndromes $+1$ to guarantee that the expected outcome of the $X_L \otimes X_L$ or $Z_L \otimes Z_L$ measurement is $+1$. Then we check the actual outcome to determine if there is a timelike logical error.

In the state injection, the correction R may differ from the actual error ϵ by $X \otimes X$ (or $Z \otimes Z$) error in the qubits

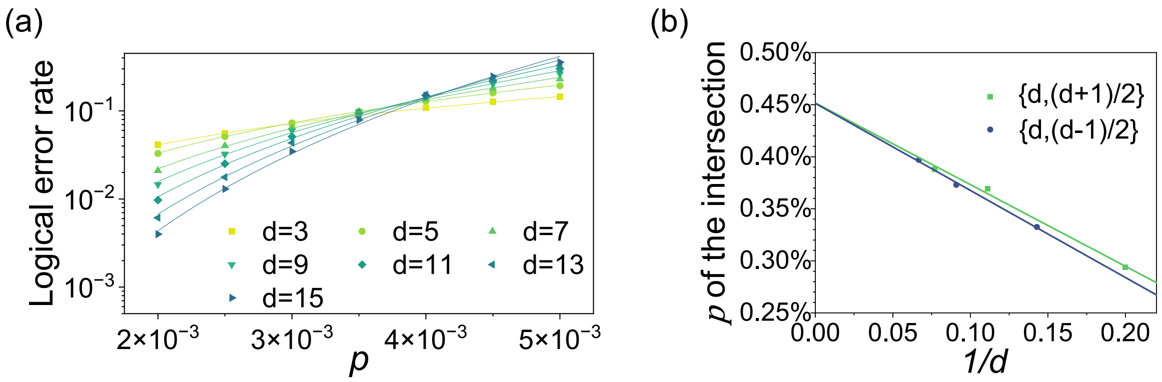


FIG. 15. (a) Logical error rates P_L of the triangular color code under the same circuit-level noise model as that in Ref. [33] with $d \leqslant 15$. (b) The intersections of pairs of curves of P_L with distances $\{d, (d+1)/2\}$ and $\{d, (d-1)/2\}$. From the linear extrapolation of the data, the threshold is around 0.45%.

of the initial Bell pairs whose syndrome corresponds to two red vertices since the measurements of red stabilizers do not have determined initial syndromes. However, they will be fault-tolerantly determined after sufficient rounds of measurements and then $X \otimes X$ will be corrected. Therefore, we do not think the decoding fails when such errors are left after decoding. From another perspective, such $X \otimes X$ (or $Z \otimes Z$) errors will not change the measurement result of the logical Pauli operators (X_L, Y_L or Z_L) since the Bell pairs and the logical Pauli operator always have an even number of common qubits.

Lastly, we list the simulation times of the numerical results in the main text. In Figs. 8(a) and 11, each point is obtained by over 10^6 simulations. In Figs. 13(a) and 13(b), we obtain each point by $10^3/p$ times of simulations, where p is the physical error rate ranging from 10^{-4} to 10^{-3}.

APPENDIX D: DIRECT COMPARISON WITH THE THRESHOLD IN REF. [33]

In the main text, we avoided directly comparing our threshold results with those in Ref. [33] because our circuit-level noise model is slightly weaker than theirs. Specifically, they assume that the measurement results (or state preparations) flip with probability p, whereas we assume $2p/3$ (the equivalent result for a depolarizing channel $\mathcal{E}_1$). Here, we provide some simulation results under the same noise model as in Ref. [33]. We tested the logical error rates of the triangular color code for $d \leqslant 15$ and evaluated the threshold using the same method as in the main text (see Fig. 15). The results show that both the logical error rates and the threshold (around 0.45%) outperform those in Ref. [33] (around 0.37%), directly demonstrating the improvements our decoding algorithm provides.

[1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. **41**, 303 (1999).

[2] R. P. Feynman, Quantum mechanical computers, Opt. News **11**, 11 (1985).

[3] M. H. Freedman, A. Kitaev, and Z. Wang, Simulation of topological field theories by quantum computers, Commun. Math. Phys. **227**, 587 (2002).

[4] J. Preskill, Reliable quantum computers, Proc. R. Soc. London, Ser. A **454**, 385 (1998).

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

[6] B. M. Terhal, Quantum error correction for quantum memories, Rev. Mod. Phys. **87**, 307 (2015).

[7] D. Gottesman, *Stabilizer Codes and Quantum Error Correction* (California Institute of Technology, 1997).

[8] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).

[9] J. K. Pachos, *Introduction to Topological Quantum Computation* (Cambridge University Press, Cambridge, 2012).

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

[11] A. M. Kubica, The ABCs of the color code: A study of topological quantum codes as toy models for fault-tolerant quantum computation and quantum phases of matter, Ph.D. thesis, California Institute of Technology, 2018.

[12] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[13] A. J. Landahl and C. Ryan-Anderson, Quantum computing by color-code lattice surgery, arXiv:1407.5103.

[14] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, Triangular color codes on trivalent graphs with flag qubits, New J. Phys. **22**, 023019 (2020).

[15] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Towards practical classical processing for the surface code, Phys. Rev. Lett. **108**, 180501 (2012).

[16] A. M. Stephens, Fault-tolerant thresholds for quantum error correction with the surface code, Phys. Rev. A **89**, 022321 (2014).

[17] A. J. Landahl, J. T. Anderson, and P. R. Rice, Fault-tolerant quantum computing with color codes, arXiv:1108.5738.

[18] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).

[19] F. Thomsen, M. S. Kesselring, S. D. Bartlett, and B. J. Brown, Low-overhead quantum computing with the color code, arXiv:2201.07806.

[20] D. Litinski and F. von Oppen, Lattice surgery with a twist: Simplifying Clifford gates of surface codes, Quantum **2**, 62 (2018).

[21] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).

[22] A. G. Fowler and C. Gidney, Low overhead quantum computation using lattice surgery, arXiv:1808.06709.

[23] C. Chamberland and E. T. Campbell, Circuit-level protocol and analysis for twist-based lattice surgery, Phys. Rev. Res. **4**, 023090 (2022).

[24] C. Gidney, In-place access to the surface code y basis, Quantum **8**, 1310 (2024).

[25] C. Chamberland and E. T. Campbell, Universal quantum computing with twist-free and temporally encoded lattice surgery, PRX Quantum **3**, 010331 (2022).

[26] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[27] S. Bravyi and J. Haah, Magic-state distillation with low overhead, Phys. Rev. A **86**, 052329 (2012).

[28] E. T. Campbell and M. Howard, Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost, Phys. Rev. A **95**, 022316 (2017).

[29] D. Litinski, Magic state distillation: Not as costly as you think, Quantum **3**, 205 (2019).

[30] Y. Li, A magic state's fidelity can be superior to the operations that created it, New J. Phys. **17**, 023037 (2015).

[31] L. Lao and B. Criger, Magic state injection on the rotated surface code, in Proceedings of the 19th ACM International Conference on Computing Frontiers (unpublished).

[32] S. Singh, A. S. Darmawan, B. J. Brown, and S. Puri, High-fidelity magic-state preparation with a biased-noise architecture, Phys. Rev. A **105**, 052410 (2022).

[33] M. E. Beverland, A. Kubica, and K. M. Svore, Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes, PRX Quantum **2**, 020341 (2021).

[34] N. Delfosse, Decoding color codes by projection onto surface codes, Phys. Rev. A **89**, 012317 (2014).

[35] J. Edmonds, Paths, trees, and flowers, Can. J. Math. **17**, 449 (1965).

[36] O. Higgott, Pymatching: A PYTHON package for decoding quantum codes with minimum-weight perfect matching, ACM Trans. Quantum Comput. **3**, 1 (2022).

[37] C. Gidney and C. Jones, New circuits and an open source decoder for the color code, arXiv:2312.08813.

[38] M. S. Kesselring, J. C. M. de la Fuente, F. Thomsen, J. Eisert, S. D. Bartlett, and B. J. Brown, Anyon condensation and the color code, PRX Quantum **5**, 010342 (2024).

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

[40] A. Steane, Multiple-particle interference and quantum error correction, Proc. R. Soc. London, Ser. A **452**, 2551 (1996).

[41] S. Bravyi, G. Smith, and J. A. Smolin, Trading classical and quantum computational resources, Phys. Rev. X **6**, 021043 (2016).

[42] P. Prabhu and C. Chamberland, New magic state distillation factories optimized by temporally encoded lattice surgery, arXiv:2210.15814.

[43] A. G. Fowler, Time-optimal quantum computation, arXiv:1210.4626.

[44] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, Phys. Rev. Lett. **102**, 110502 (2009).

[45] R. Raussendorf, J. Harrington, and K. Goyal, A fault-tolerant one-way quantum computer, Ann. Phys. **321**, 2242 (2006).

[46] T. Jochym-O'Connor and S. D. Bartlett, Stacked codes: Universal fault-tolerant quantum computation in a two-dimensional layout, Phys. Rev. A **93**, 022323 (2016).

[47] B. J. Brown, A fault-tolerant non-Clifford gate for the surface code in two dimensions, Sci. Adv. **6**, eaay4929 (2020).

[48] C. Chamberland and A. W. Cross, Fault-tolerant magic state preparation with flag qubits, Quantum **3**, 143 (2019).

[49] C. Chamberland and K. Noh, Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits, npj Quantum Inf. **6**, 91 (2020).

[50] S. Koutsioumpas, D. Banfield, and A. Kay, The smallest code with transversal T, arXiv:2210.14066.

[51] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, Phys. Rev. A **83**, 020302(R) (2011).

[52] A. G. Fowler, A. C. Whiteside, A. L. McInnes, and A. Rabbani, Topological code AUTOTUNE, Phys. Rev. X **2**, 041003 (2012).

[53] O. Higgott, T. C. Bohdanowicz, A. Kubica, S. T. Flammia, and E. T. Campbell, Improved decoding of circuit noise and fragile boundaries of tailored surface codes, Phys. Rev. X **13**, 031007 (2023).

[54] B. Criger and I. Ashraf, Multi-path summation for decoding 2D topological codes, Quantum **2**, 102 (2018).

[55] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A **70**, 052328 (2004).

[56] K. Sahay and B. J. Brown, Decoder for the triangular color code by matching on a möbius strip, PRX Quantum **3**, 010310 (2022).

[57] S.-H. Lee, A. Li, and S. D. Bartlett, Color code decoder with improved scaling for correcting circuit-level noise, arXiv:2404.07482.

[58] D. Gottesman, The Heisenberg representation of quantum computers, in *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, edited by S. P. Corney, R. Delbourgo, and P. D. Jarvis (International Press, Cambridge, MA, 1999), pp. 32–43.