Achieving Fault Tolerance on Capped Color Codes with Few Ancillas

Theerapat Tansuwannont[1,2,*] and Debbie Leung[3,4,†]

[1] *Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

[2] *Duke Quantum Center and Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, USA*

[3] *Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

[4] *Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada*

(Received 12 December 2021; revised 27 May 2022; accepted 21 June 2022; published 11 August 2022)

Attaining fault tolerance while maintaining low overhead is one of the main challenges in a practical implementation of quantum circuits. One major technique that can overcome this problem is the flag technique, in which high-weight errors arising from a few faults can be detected by a few ancillas and distinguished using subsequent syndrome measurements. The technique can be further improved using the fact that, for some families of codes, errors of any weight are logically equivalent if they have the same syndrome and weight parity, as shown in our previous work [Tansuwannont and Leung, Phys. Rev. A 104, 042410 (2021)]. In this work, we develop a notion of distinguishable fault set that captures both concepts of flags and weight parities, and extend the use of weight parities in error correction to families of capped and recursive capped color codes. We also develop fault-tolerant protocols for error correction, measurement, state preparation, and logical T-gate implementation via code switching, which are sufficient for performing fault-tolerant Clifford computation on a capped color code, and performing fault-tolerant universal quantum computation on a recursive capped color code. Our protocols for a capped or a recursive capped color code of any distance require only two ancillas, assuming that the ancillas can be reused. The concept of distinguishable fault set also leads to a generalization of the definitions of fault-tolerant gadgets proposed by Aliferis, Gottesman, and Preskill.

DOI: 10.1103/PRXQuantum.3.030322

I. INTRODUCTION

Fault-tolerant error correction (FTEC), a procedure that suppresses error propagation in a quantum circuit, is one of the most important components for building large-scale quantum computers. Given that the physical error rate is below some constant threshold value, a FTEC scheme along with other schemes for fault-tolerant quantum computation (FTQC) allow us to fault-tolerantly simulate any quantum circuit with arbitrarily low logical error rates [2–10]. However, a lower logical error rate requires more overhead (e.g., quantum gates and ancilla qubits) [11–14]. Therefore, fault-tolerant protocols that require a small number of ancillas and give a high threshold value are very desirable for practical implementation.

Traditional FTEC schemes require a substantial number of ancillas for error syndrome measurements. For example, the Shor error-correction (EC) scheme [2,15], which is applicable to any stabilizer code, requires as many ancillas as the maximum weight of the stabilizer generators. The Knill EC scheme [16], which is also applicable to any stabilizer code, requires two code blocks of ancillas. Meanwhile, the Steane EC scheme [17,18], which is applicable to any Calderbank-Shor-Steane (CSS) code, requires one code block of ancillas. (The Shor scheme also requires repeated syndrome measurement, while the Knill and the Steane schemes do not.) There are several recently proposed schemes that require fewer ancillas. Yoder and Kim proposed a FTEC scheme for the $[[7,1,3]]$ code that requires only two ancillas [19], and their scheme is further developed into a well-known flag FTEC scheme for the $[[5,1,3]]$ code and the $[[7,1,3]]$ code that also require only two ancillas [20] (where an $[[n,k,d]]$ stabilizer code encodes k logical qubits into n physical qubits and has distance d). In general, a flag FTEC scheme for

*t.tansuwannont@duke.edu

†wcleung@uwaterloo.ca

any stabilizer code requires as few as $d+1$ ancillas, where d is the code distance [21], with further reduction known for certain families of codes [20,22–25]. The flag technique can also be applied to other schemes for FTQC [26–35].

How errors spread during the protocols depends on several factors, such as the order of quantum gates in the circuits for syndrome measurement and the choice of stabilizer generators being measured. The idea behind the flag technique is that a few ancillas are added to the circuits in order to detect errors of high weight arising from a few faults, and the errors will be distinguished by their syndromes obtained from subsequent syndrome measurements. Note that some possible errors may be logically equivalent and need not be distinguished, and for some families of codes, we can tell whether the errors are logically equivalent using their syndromes and error weight parities. Tansuwannont and Leung [1] combines the ideas of flags and weight parities to construct a FTEC scheme for a [[49, 1, 9]] concatenated Steane code, which can correct up to three faults and requires only two ancillas. In such a scheme, the weight parity of the error in each sub-block, which is the lower-level [[7, 1, 3]] code, is determined by the results from measuring the generators of the higher-level [[7, 1, 3]] code. The scheme in Ref. [1] uses very few ancillas compared to conventional schemes for a concatenated code (which is constructed by replacing a physical qubit by a code block) and is expected to be applicable to concatenated codes other than the [[49, 1, 9]] code.

There are families of codes that attain high distance without code concatenation. Topological codes in which the code distance can be made arbitrarily large by increasing the lattice size are good candidates for practical implementation of quantum computers since fault-tolerant protocols for these codes typically give very high accuracy thresholds [36–52]. Examples of two-dimensional (2D) topological stabilizer codes are 2D toric codes [4,53] and 2D color codes [54]. These codes are suitable for physical implementations using superconducting qubits [24,25,55] and qubits realized by Majorana zero modes [56,57] since qubits can be arranged on a 2D plane and only quantum gates involving neighboring qubits are required. Toric codes and color codes can be transformed to one another using the techniques developed in Ref. [58] (see also Ref. [59]).

The simplest way to perform FTQC on a topological stabilizer code is to implement logical gates by applying physical gates transversally since doing so does not spread errors (therefore fault tolerant). Unfortunately, it is known by the Eastin-Knill theorem that a universal set of quantum operations cannot be achieved using only transversal gates [60]. Moreover, logical gates that can be implemented transversally on a 2D topological stabilizer code are in the Clifford group [61] (see also Ref. [62]). The Clifford group can be generated by the Hadamard gate (H), the $\pi/4$ gate (S), and the CNOT gate [63,64]. A transversal CNOT gate is achievable by both 2D toric codes and 2D color codes since these codes are in the CSS code family [65,66]. In addition, the 2D color codes have transversal H and S gates [54], so any Clifford operation can be implemented transversally on any 2D color code.

Implementing only Clifford gates on a 2D color code is not particularly interesting since a Clifford operation can be efficiently simulated by a classical computer (the result is known as the Gottesman-Knill theorem) [67,68]. However, universality can be achieved by Clifford gates together with any gate not in the Clifford group [69]. There are two compelling approaches for implementing a non-Clifford gate on a 2D color code: magic state distillation [70] and code switching [71–74]. The former approach focuses on producing high-fidelity T states from noisy T states and Clifford operations, where $|T\rangle = (|0\rangle + \sqrt{i}|1\rangle)/\sqrt{2}$ is the state that can be used to implement the non-Clifford $T = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{i} \end{pmatrix}$ operation. By replacing any physical gates and qubits with logical gates and blocks of code, a logical T gate can be implemented using a method similar to that proposed in Ref. [70]. The latter approach uses the gauge fixing method to switch between a 2D color code (in which Clifford gates are transversal) and a 3D color code (in which the T gate is transversal). A recent study [75] that compares the overhead required for these two approaches shows that code switching does not outperform magic state distillation when certain FT schemes are used, except for some small values of the physical error rate. Nevertheless, their results do not rule out the possibilities of FT schemes that have yet to be discovered, which the authors are hopeful could reduce the overhead required for either of the aforementioned approaches.

The EC technique using weight parities introduced in Ref. [1] was originally developed for the [[49,1,9]] code obtained from concatenating the [[7, 1, 3]] codes. The [[7, 1, 3]] code is also the smallest 2D color code. Surprisingly, we find that 2D color codes of any distance have certain properties that make similar techniques applicable, under appropriate modifications of the original code to be described in this paper. In order to obtain the weight parity of an error on a 2D color code, we need to make measurements of stabilizer generators of a bigger code that contains the 2D color code as a subcode. In contrast to Ref. [1], the bigger code in this work is not obtained from code concatenation. Our development for FTEC protocols leads to a family of capped color codes, which are CSS subsystem codes [76,77]. We study two stabilizer codes obtained from a (subsystem) capped color code through gauge fixing, namely capped color codes in H form and T form. The code in H form that contains a 2D color code as a subcode has transversal Clifford gates, while the code in T form has transversal CNOT and transversal T gates. In fact, our capped color codes bear similarities to the subsystem codes presented in Refs. [78–80], in which qubits can be arranged on a 2D plane. In this work, we focus mainly on the construction of circuits for measuring generators of a capped color code in H form, and the construction of a FTEC scheme as well as other fault-tolerant schemes for measurement, state preparation, and Clifford operation. We also prove that our fault-tolerant schemes for capped color codes in H form of *any distance* require only two ancillas (assuming that the ancillas can be reused). In addition, we construct a family of recursive capped color codes by recursively encoding the top qubit of capped color codes. Circuits for measuring generators of capped color codes in H form also work for recursive capped color codes, so fault-tolerant Clifford computation on a recursive capped color code of any distance using only two ancillas is possible. We also show that a logical T gate can be fault-tolerantly implemented on a recursive capped color code of any distance using only two ancillas via code switching, leading to a complete set of operations for fault-tolerant universal quantum computation.

This paper is organized as follows. In Sec. II, we provide a brief review of the EC technique using flags and error weight parities. We also develop a notion of distinguishable fault set in Definition 3, which is the central idea of this work. In Sec. III, we review basic properties of the 3D color code of distance 3 (which is defined as a subsystem code). We then provide a construction of circuits for measuring the stabilizer generators of the 3D color code in H form that give a distinguishable fault set. In Sec. IV, we define families of capped and recursive capped color codes, whose properties are very similar to those of the 3D color code of distance 3. Afterwards, circuits for measuring the stabilizer generators of the capped color code in H form are constructed using ideas from the previous section. We prove Theorem 1 that states sufficient conditions for the circuits that can give a distinguishable fault set, then prove Theorems 2 and 3 that state that, for a capped color code in H form of any distance, a distinguishable fault set can be obtained if the circuits for measuring generators are flag circuits of a particular form. The circuits that work for capped color codes are also applicable to recursive capped color codes. In Sec. V, we discuss an alternative version of fault-tolerant gadgets whose definitions are modified so that they are compatible with the notion of distinguishable fault set. Afterwards, we construct fault-tolerant protocols for capped and recursive capped color codes in H form. Some protocols described in this work are also applicable to other stabilizer codes whose generator measurement circuits give a distinguishable fault set. Last, we discuss our results and provide directions for future work in Sec. VI.

II. FLAGS AND ERROR WEIGHT PARITIES IN ERROR CORRECTION

In this section, we start by providing a brief review of the flag EC technique applied to the case of one fault in Sec. II A. Next, we extend the idea to the case of multiple faults in Sec. II B and introduce a notion of distinguishable fault set in Definition 3. Afterwards, we explain how weight parities can be used in error correction in Sec. II C. The equivalence of Pauli errors with the same syndrome and weight parity proved for the $[[7,1,3]]$ Steane code in Ref. [1] is also extended to a bigger family of codes in Lemma 1.

A. Flag error correction

Quantum computation is prone to noise, and an error on a few qubits can spread and cause a big problem in the computation if the error is not treated properly. One way to protect quantum data against noise is to use a quantum error correcting code (QECC) to encode a small number of logical qubits into a larger number of physical qubits. A quantum $[[n,k,d]]$ stabilizer code [67,81] encodes k logical qubits into n physical qubits and can correct errors up to weight $\tau = \lfloor (d-1)/2 \rfloor$. Quantum error correction (QEC) is a process that aims to undo the corruption that happens to a codeword.

A stabilizer code is a simultaneous $+1$ eigenspace of a list of commuting independent Pauli operators; they generate the stabilizer group for the code. For a stabilizer code, the EC procedure involves measurements of stabilizer generators, which results in an error syndrome. The QEC is designed so that the more likely Pauli errors are either logically equivalent or have distinguishable syndrome. If the weight of the Pauli error E occurred to a codeword is no bigger than τ, E can be identified by the error syndrome $\vec{s}(E)$ obtained from the generator measurements, and can be corrected by applying $E^\dagger$ to the codeword.

The above working principle for a stabilizer code assumes that the syndrome measurements are perfect. In practice, every step in a quantum computation, including those in the syndrome measurements, is subject to error. An initial error can lead to a complex overall effect in the circuit. We adhere to the following terminologies and noise model in our discussion.

Definition 1 (Location, noise model, and fault [10]): A circuit consists of a number of time steps and a number of qubits and is specified by operations to the qubits in each time step. The operations can be single-qubit state preparation, one- or two-qubit gates, or single-qubit measurement. (When nothing happens to a qubit, it goes through the one-qubit gate of identity.) A *location* is labeled by a time step and the index (or indices) of a qubit (or pair of qubits) involved in an operation.

We consider the circuit-level noise in which every location is followed by *depolarizing noise*: every one-qubit operation is followed by a single-qubit Pauli error I, X, Y, or Z, and every two-qubit operation is followed by a two-qubit Pauli error of the form $P_1 \otimes P_2$, where $P_1, P_2 \in \{I, X, Y, Z\}$. For a single qubit measurement (which outputs a classical bit of information), the operation is followed by either no error or a bit-flip error; this is equivalent to having a single-qubit X (or Z) error before a measurement in the Z (or X) basis.

A *fault* is specified by a location and a nontrivial one- or two-qubit Pauli operation that describes a deviation from the ideal operation on the location. This Pauli operation is called the "Pauli error due to the fault".

A small number of faults during the measurements can lead to an error of weight higher than τ that may cause the EC protocol to fail. To see this, first, we describe how an error of weight 1 or 2 arising from a faulty operation can propagate through a circuit and become an error of higher weight. Specifically, a Hadamard gate and a CNOT gate will transform X-type and Z-type errors as

$$\begin{array}{rll} H: & X \mapsto Z, & Z \mapsto X, \\ \text{CNOT}: & XI \mapsto XX, & ZI \mapsto ZI, \\ & IX \mapsto IX, & IZ \mapsto ZZ. \end{array}$$

To see how errors from a few faults can cause an EC protocol to fail, let us consider a circuit for measuring a stabilizer generator of the Steane code as an example. The [[7, 1, 3]] Steane code [66] is a stabilizer code that can be described by the generators

$$\begin{array}{lccccccclccccccc} g_1^x: & I & I & I & X & X & X & X, & g_1^z: & I & I & I & Z & Z & Z & Z, \\ g_2^x: & I & X & X & I & I & X & X, & g_2^z: & I & Z & Z & I & I & Z & Z, \\ g_3^x: & X & I & X & I & X & I & X, & g_3^z: & Z & I & Z & I & Z & I & Z. \end{array}$$

Logical X and logical Z operators of the Steane code are $X^{\otimes 7}M$ and $Z^{\otimes 7}N$ for any stabilizers M, N. The syndrome is a 6-bit string of the form $(\vec{s}_x|\vec{s}_z)$, with the ith bit being 0 (or 1) if measuring the ith generator (ordered as g_1^x, g_2^x, g_3^x, then g_1^z, g_2^z, g_3^z) gives the $+1$ (or -1) eigenvalue.

Suppose that during the syndrome measurement all circuits for measuring stabilizer generators are perfect except for a circuit for measuring g_1^z that has at most one fault. Consider a circuit for measuring g_1^z and storing the syndrome using one ancilla qubit (called the *syndrome ancilla*) as in Fig. 1(a). Also, assume that at most one CNOT gate causes either an II, IZ, ZI, or ZZ error. Because of error propagation, a Z error occurred to the syndrome ancilla can propagate back to one or more data qubits. As a result, we find that possible errors on data qubits arising from at most one CNOT fault (up to multiplication of g_1^z) are

$$I, Z_4, Z_5, Z_6, Z_7, Z_6Z_7. \tag{1}$$

A circuit fault may also cause the syndrome bit to flip. In order to obtain the syndrome exactly corresponding to the data error, one can perform full syndrome measurements until the outcomes are repeated two times in a row, then do the error correction using the repeated syndrome. However, note that the Steane code that can correct any error up to weight 1 must be able to correct the following errors as well:

$$I, Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7. \tag{2}$$

Errors Z_1 and Z_6Z_7 have the same syndrome $(0, 0, 1|0, 0, 0)$ but are not logically equivalent, and subsequent syndrome measurements cannot distinguish between these two cases. This means that if a CNOT fault leads to the Z_6Z_7 error, a correction step for the syndrome $(0, 0, 1|0, 0, 0)$ that applies $Z_1^\dagger$ to the data qubits will result in a logical error $Z_1Z_6Z_7$ on the data qubits, causing the EC protocol to fail.

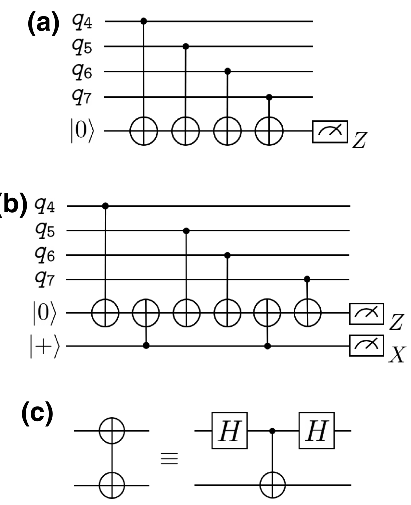


FIG. 1. (a) An example of a nonflag circuit for measuring generator g_1^z of the [[7, 1, 3]] code. Only qubits on which the operator acts are displayed. The measurement results 0 and 1 obtained from the syndrome ancilla correspond to the $+1$ and -1 eigenvalues of g_1^z. (b) An example of a flag circuit for measuring g_1^z. The state of the flag ancilla can flip from $|+\rangle$ to $|-\rangle$ if some fault occurs in between two flag CNOT gates. A circuit for measuring an X-type generator can be obtained by replacing each CNOT gate with the gate shown in (c).

The goal of this work is to design an EC protocol that is *fault tolerant*; that is, we want to make sure that any subsequent error arising from a small number of faults will still be correctable by the protocol regardless of its weight (the formal definitions of fault tolerance will be discussed in Sec. V A).

One way to solve the error distinguishing issue is to use traditional FTEC schemes such as those proposed by Shor [2,15], Steane [17,18], or Knill [16]. However, these schemes require a large number of ancillas. An alternative way to solve the problem is to add an additional ancilla qubit in a circuit for measuring g_1^z, as shown in Fig. 1(b). A circuit of this form is called a *flag circuit* [20] [in contrast to the circuit in Fig. 1(a), which is called a *nonflag circuit*]. The additional ancilla qubit is called a *flag ancilla*, which is initially prepared in the state $|+\rangle$. There are two types of CNOT gates in a flag circuit: a *data* CNOT gate that couples one of the data qubits and the syndrome ancilla, and a *flag* CNOT gate that couples the flag ancilla and the syndrome ancilla. Whenever a data CNOT gate in between two flag CNOT gates causes either an IZ or ZZ error, a Z error will propagate from the syndrome ancilla to the flag ancilla, causing the state of the flag ancilla to flip to $|-\rangle$. In general, a flag circuit may have more than one flag ancilla, and data and flag CNOT gates may be arranged in a complicated way so that a certain number of faults can be caught by the flag ancillas.

By using the circuit in Fig. 1(b) for measuring g_1^z, we find that possible errors on the data qubits arising from at most one CNOT fault corresponding to each flag measurement outcome are

$$\begin{aligned} 0: &\quad I, Z_4, Z_5, Z_6, Z_7, \\ 1: &\quad I, Z_4, Z_6Z_7, Z_7, \end{aligned} \tag{3}$$

where the outcomes 0 and 1 correspond to $|+\rangle$ and $|-\rangle$ states, respectively. We can see that the flag measurement outcome is 1 whenever Z_6Z_7 occurs. In contrast, an input error Z_1 will not flip the state of the flag ancilla, so it always corresponds to the flag measurement outcome 0. Therefore, Z_1 and Z_6Z_7 can be distinguished using the flag measurement outcome, and an appropriate error correction for each case can be applied to correct such an error. The main advantage of the flag technique is that the number of ancillas required for the flag FTEC protocol is relatively small compared to that required for the traditional FTEC protocols (assuming that ancilla preparation and measurement are fast and the ancillas can be reused).

B. Distinguishable fault set

For a general stabilizer code that can correct errors up to weight $\tau = \lfloor (d-1)/2 \rfloor$, we would like to construct circuits for syndrome measurement in a way that all possible errors arising from up to t faults (where $t \leq \tau$) can be corrected, and t is as close to τ as possible. Note that these errors include any single-qubit errors and errors arising from any fault in any circuit involved in the syndrome measurement. For simplicity, this work will focus mainly on a stabilizer code in the CSS code family [65,66], in which X-type and Z-type errors can be detected and corrected separately.

For a given CSS code, a circuit for measuring a Z-type generator will look similar to the circuit in Fig. 1(a) or 1(b), except that there will be w data CNOT gates for a Z-type generator of weight w. A circuit can have any number of flag ancillas (or have no flag ancillas). There are several factors that can determine the ability to distinguish possible errors; for example, the number of flag ancillas, the ordering of data and flag CNOT gates, and the choice of generators being used for the syndrome measurement [20]. A circuit for measuring an X-type generator is similar to a circuit for measuring a Z-type generator, except that each CNOT gate is replaced by the gate displayed in Fig. 1(c).

For a given t, finding all possible combinations of faults up to t faults can be laborious since there are many circuits involved in the syndrome measurement, and each circuit has many gates. To simplify our analysis, we first consider the case that there is only one CNOT fault in one of the circuits for measuring Z-type generators [similar to Fig. 1(a) or 1(b)]. Suppose that there are a total of c flag ancillas involved in a single round of the full syndrome measurement (counted from all circuits). We define a *flag vector* $\in \mathbb{Z}_2^c$ to be a bitstring wherein each bit is the measurement outcome of each flag ancilla. There are two mathematical objects associated with each fault: a data error arising from the fault and a flag vector corresponding to the fault.

Recall that a faulty CNOT gate can cause a two-qubit error of the form $P_1 \otimes P_2$, where $P_1, P_2 \in \{I, X, Y, Z\}$. However, there are many cases of a single fault that are equivalent, meaning that they can give rise to the same data error and the same flag vector. We find that all possible cases in which a single fault can lead to a purely Z-type error on the data qubits can be obtained by considering only (1) the cases that a faulty CNOT gate in a circuit for measuring a Z-type generator causes an IZ error, and (2) the cases that a Z error occurs to any data or ancilla qubit. This follows from the following facts [23].

1. The case that a faulty CNOT gate causes a ZZ error is equivalent to the case that the preceding CNOT gate causes an IZ error (while the case that the first CNOT gate in a circuit causes a ZZ error is equivalent to the case that a Z error occurs to an ancilla qubit).
2. The case that a faulty CNOT gate causes an XZ or YZ error is equivalent to the case that an X error occurs to a data qubit and a faulty CNOT gate causes an IZ or ZZ error.
3. The case that a faulty CNOT gate causes an XI, YI, ZI, IX, XX, YX, or ZX error can be considered as the case that a single-qubit error occurs to a

data qubit since an X error occurred to the syndrome ancilla will not propagate back to any data qubit.

4. The case that a faulty CNOT gate causes an IY, XY, YY, or ZY error is similar to the case that a faulty CNOT gate causes an IZ, XZ, YZ, or ZZ error.
5. An ancilla preparation or measurement fault can be considered as the case that an X or Z error occurred to an ancilla qubit (either syndrome or flag ancilla).
6. A CSS code can detect and correct X-type and Z-type errors separately, and a single fault in a circuit for measuring an X-type generator cannot cause a Z-type error of weight greater than 1 (and vice versa).

Moreover, if X-type and Z-type generators have similar forms and the gate permutations in the measuring circuits are the same, then all possible faults that can lead to X-type errors on the data qubits are of similar form.

If there are many faults during the protocol, the data errors and the flag vectors caused by each fault can be combined [1]. In particular, a fault combination can be defined as follows.

Definition 2 (**Fault combination**): A *fault combination* $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ is a set of r faults $\lambda_1, \lambda_2, \ldots, \lambda_r$. Suppose that the Pauli error due to fault λ_i can propagate through the circuit and lead to *data error* E_i and *flag vector* $\vec{f}_i$. The *combined data error* $\mathbf{E}$ and *cumulative flag vector* $\vec{\mathbf{f}}$ corresponding to Λ are defined as

$$\mathbf{E} = \prod_{i=1}^{r} E_i, \tag{4}$$

$$\vec{\mathbf{f}} = \sum_{i=1}^{r} \vec{f}_i \pmod 2. \tag{5}$$

Note that the error syndrome of the combined data error is $\vec{s}(\mathbf{E}) = \sum_{i=1}^{r} \vec{s}(E_i) \pmod 2$. For example, suppose that a fault combination Λ arises from two faults λ_1 and λ_2 that can lead to data errors E_1 and E_2, and cumulative flag vectors $\vec{f}_1$ and $\vec{f}_2$. Then, the combined data error $\mathbf{E}$ and the cumulative flag vector $\vec{\mathbf{f}}$ of Λ are $\mathbf{E} = E_1 \cdot E_2$ and $\vec{\mathbf{f}} = \vec{f}_1 + \vec{f}_2 \pmod 2$.

When faults occur in an actual protocol, the faulty locations and the combined data error are not known. In order to determine the combined data error so that the error correction can be done, we try to measure the error syndrome of the combined data error, and calculate the cumulative flag vector from the flag measurement results obtained since the beginning of the protocol. These measurements, in turn, are subject to errors. The full syndrome measurements will be performed until the syndromes and the cumulative flag vectors are repeated a certain number of times (similar to the Shor FTEC scheme); the full details of the protocol will be described in Sec. V B. (Note that, by defining the cumulative flag vector as a sum of flag vectors, we lose the information about the ordering in which each fault occurs. However, we find that fault-tolerant protocols presented in this work can still be constructed without such information.)

As previously explained, error correction can fail if there are different faults that lead to nonequivalent errors, but there is no way to distinguish them using their error syndromes or flag measurement results. To avoid this, all possible fault combinations must satisfy some conditions so that they can be distinguished. In particular, for a given set of circuits for measuring stabilizer generators, all possible fault combinations can be found, and their corresponding combined data error and cumulative flag vector can be calculated. Let the fault set $\mathcal{F}_t$ be the set of all possible fault combinations arising from up to t faults. We will be able to distinguish all fault combinations if the fault set satisfies the conditions in the following definition.

Definition 3 (**Distinguishable fault set**): Let the *fault set* $\mathcal{F}_t$ denote the set of all possible fault combinations arising from up to t faults, and let S be the stabilizer group of the quantum error correcting code used to encode the data. We say that $\mathcal{F}_t$ is *distinguishable* if, for any pair of fault combinations $\Lambda_p, \Lambda_q \in \mathcal{F}_t$, at least one of the following conditions is satisfied:

1. $\vec{s}(\mathbf{E}_p) \neq \vec{s}(\mathbf{E}_q)$,
2. $\vec{\mathbf{f}}_p \neq \vec{\mathbf{f}}_q$,
3. $\mathbf{E}_p = \mathbf{E}_q \cdot M$ for some stabilizer $M \in S$,

where $\mathbf{E}_p, \vec{\mathbf{f}}_p$ correspond to Λ_p, and $\mathbf{E}_q, \vec{\mathbf{f}}_q$ correspond to Λ_q. Otherwise, we say that $\mathcal{F}_t$ is *indistinguishable.*

An example of a distinguishable fault set with $t = 1$ is the fault set corresponding to Eq. (3) (assuming that a fault occurs in a circuit for measuring g_1^z only). In that case, we can see that, for any pair of faults, either the syndromes of the data errors or the flag measurement outcomes are different.

The following proposition states the relationship between "correctable" and "detectable" faults. This is similar to the fact that a stabilizer code of distance d can detect errors up to weight $d-1$ and can correct errors up to weight $\tau = \lfloor (d-1)/2 \rfloor$ [67].

Proposition 1. *$\mathcal{F}_t$ is distinguishable if and only if a fault combination corresponding to a nontrivial logical operator and the zero cumulative flag vector is not in $\mathcal{F}_{2t}$.*

Proof. ($\Rightarrow$) Let $\Lambda_p, \Lambda_q \in \mathcal{F}_t$ be fault combinations arising from up to t faults, let $\tilde{\Lambda}_r \in \mathcal{F}_{2t}$ be a fault combination arising from up to $2t$ faults, and let S be the stabilizer group. First, observe that, for any $\tilde{\Lambda}_r \in \mathcal{F}_{2t}$, there exist $\Lambda_p, \Lambda_q \in \mathcal{F}_t$ such that $\tilde{\Lambda}_r = \Lambda_p \cup \Lambda_q$ (where the union of two fault combinations is similar to the union of two sets). Now suppose that $\mathcal{F}_t$ is distinguishable. Then, for each pair of

Λ_p, Λ_q in $\mathcal{F}_t$, $\vec{s}(\mathbf{E}_p) \neq \vec{s}(\mathbf{E}_q)$ or $\vec{\mathbf{f}}_p \neq \vec{\mathbf{f}}_q$ or $\mathbf{E}_p = \mathbf{E}_q \cdot M$ for some stabilizer $M \in S$. We find that $\tilde{\Lambda}_r = \Lambda_p \cup \Lambda_q$ corresponds to $\mathbf{E}_r$ and $\vec{\mathbf{f}}_r$ such that $\vec{s}(\mathbf{E}_r) = \vec{s}(\mathbf{E}_p) + \vec{s}(\mathbf{E}_q) \neq 0$ or $\vec{\mathbf{f}}_r = \vec{\mathbf{f}}_p + \vec{\mathbf{f}}_q \neq 0$ or $\mathbf{E}_r = \mathbf{E}_p \cdot \mathbf{E}_q = M$ for some stabilizer $M \in S$. This is true for any $\tilde{\Lambda}_r \in \mathcal{F}_{2t}$, meaning that there is no fault combination in $\mathcal{F}_{2t}$ that corresponds to a nontrivial logical operator and the zero cumulative flag vector.

($\Leftarrow$) As before, we know that, for any $\tilde{\Lambda}_r \in \mathcal{F}_{2t}$, there exist $\Lambda_p, \Lambda_q \in \mathcal{F}_t$ such that $\tilde{\Lambda}_r = \Lambda_p \cup \Lambda_q$. Now suppose that $\mathcal{F}_t$ is indistinguishable. Then, there are some pairs of Λ_p, Λ_q in $\mathcal{F}_t$ such that $\vec{s}(\mathbf{E}_p) = \vec{s}(\mathbf{E}_q)$, $\vec{\mathbf{f}}_p = \vec{\mathbf{f}}_q$, and $\mathbf{E}_p \cdot \mathbf{E}_q$ is not a stabilizer in S. For such pairs, we find that $\tilde{\Lambda}_r = \Lambda_p \cup \Lambda_q$ corresponds to $\mathbf{E}_r$ and $\vec{\mathbf{f}}_r$ such that $\vec{s}(\mathbf{E}_r) = \vec{s}(\mathbf{E}_p) + \vec{s}(\mathbf{E}_q) = 0$, $\vec{\mathbf{f}}_r = \vec{\mathbf{f}}_p + \vec{\mathbf{f}}_q = 0$, and $\mathbf{E}_r = \mathbf{E}_p \cdot \mathbf{E}_q$ is not a stabilizer in S. Therefore, there is a fault combination corresponding to a nontrivial logical operator and the zero cumulative flag vector in $\mathcal{F}_{2t}$. ■

Finding a circuit configuration that gives a distinguishable fault set is one of the main goals of this work. We claim that, for a given set of circuits for measuring generators of a stabilizer code, if the fault set is distinguishable, a FTEC protocol for such a code can be constructed. However, we defer the proof of this claim until Sec. V B.

C. Finding equivalent errors using error weight parities

One goal of this work is to find a good combination of stabilizer code and a set of circuits for measuring the code generators in which the corresponding fault set is distinguishable. As we see in Definition 3, whether each pair of fault combinations can be distinguished depends on the syndrome of the combined data error and the cumulative flag vector corresponding to each fault combination, and these features heavily depend on the structure of the circuits. However, we should note that there is no need to distinguish a pair of fault combinations whose combined data errors are logically equivalent. Therefore, if the circuits for a particular code are designed in a way that large portions of fault combinations can give equivalent errors, the fault set arising from the circuits will be more likely distinguishable.

For a general stabilizer code, it is not obvious to see whether two Pauli errors with the same syndrome are logically equivalent or off by a multiplication of some nontrivial logical operator. Fortunately, for some CSS codes, it is possible to check whether two Pauli errors with the same syndrome are logically equivalent by comparing their error weight parities, defined as follows.

Definition 4 (Weight parity): The *weight parity* of Pauli error E, denoted by $\mathrm{WP}(E)$, is 0 if E has even weight, and it is 1 if E has odd weight.

In Ref. [1], we proved that, for the [[7, 1, 3]] Steane code and the [[23, 1, 7]] Golay code, errors with the same syndrome and weight parity are logically equivalent. In this work, the idea is further extended to a family of $[[n, k, d]]$ CSS codes in which n is odd, k is 1, all stabilizer generators have even weight, and $X^{\otimes n}$ and $Z^{\otimes n}$ are logical X and logical Z operators, respectively. The lemma (adapted from Claim 1 of Ref. [1]) is as follows.

Lemma 1. *Let C be an $[[n, k, d]]$ CSS code in which n is odd, $k = 1$, all stabilizer generators have even weight, and $X^{\otimes n}$ and $Z^{\otimes n}$ are logical X and logical Z operators. Also, let S_x, S_z be subgroups generated by X-type and Z-type generators of C, respectively, and let E_1, E_2 be Pauli errors of any weights with the same syndrome.*

1. *Suppose that E_1, E_2 are Z-type errors. Then E_1, E_2 have the same weight parity if and only if $E_1 = E_2 \cdot M$ for some $M \in S_z$.*
2. *Suppose that E_1, E_2 are X-type errors. Then E_1, E_2 have the same weight parity if and only if $E_1 = E_2 \cdot M$ for some $M \in S_x$.*

Proof. We focus on the first case when E_1, E_2 are Z-type errors and omit the similar proof for the second case. First, recall that the normalizer group of the stabilizer group (the subgroup of Pauli operators that commute with all stabilizers) is generated by the stabilizer generators together with the logical X and the logical Z. Since E_1, E_2 have the same syndrome, their product $N = E_1E_2$ has trivial syndrome, and is thus in the normalizer group. So we can express N as a product of the stabilizer generators and the logical X and Z. But there is no X-type factors (since N is Z type). Therefore, $N = M(Z^{\otimes n})^a$, where $M \in S_z$ and $a \in \{0, 1\}$.

Next, we make an observation. Let M_1, M_2 be two Z-type operators, with respective weights w_1, w_2. The weight of the product M_1M_2 is $w_1 + w_2 - 2c$, where c is the number of qubits supported on both M_1 and M_2. From this observation, and the fact that all generators have even weight, we know that M has even weight. Also, from the same observation, and the hypothesis that E_1, E_2 have the same weight parity, N also has even weight. If $a = 1$, $N = M(Z^{\otimes n})^a$ will contradict the observation, so $a = 0$, $N = M$, and $E_1E_2 = M \in S_z$ as claimed. On the other hand, if we assume that E_1, E_2 have different weight parities, then N has odd weight and $a = 1$, which implies that $E_1E_2 = M(Z^{\otimes n})$ for some $M \in S_z$. ■

Lemma 1 provides a possible way to perform error correction using syndromes and weight parities, and it can help us find a good code and circuits in which the fault set is distinguishable. In particular, for a given CSS code satisfying Lemma 1, if the error syndrome and the weight parity of the data error can be measured perfectly, then an EC operator that can map the erroneous codeword back to the original codeword can be determined without failure. The EC operator can be any Pauli operator that has the same syndrome and the same weight parity as those of the data error. For example, if the [[7, 1, 3]] Steane code is being used and the data error is $Z_1Z_3Z_6Z_7$, we can use Z_1Z_2 as an EC operator to do the error correction.

However, measuring the weight parity should not be done directly on the codeword; measuring weight parities of Z-type and X-type errors correspond to measuring $X^{\otimes n}$ and $Z^{\otimes n}$, respectively, which may destroy the superposition of the encoded state. Moreover, $X^{\otimes n}$ and $Z^{\otimes n}$ do not commute. Fortunately, if we have two codes C_1, C_2 such that C_1 is a subcode of C_2, then the weight parity of an error on C_1 can sometimes be determined by the measurement results of the generators of C_2.

In Ref. [1], in which a FTEC protocol for a [[49, 1, 9]] concatenated Steane code is developed, we considered the case that C_1 is the [[7, 1, 3]] Steane code and C_2 is the [[49, 1, 9]] concatenated code. The error weight parities for each sub-block of the seven-qubit code were determined by the syndrome obtained from the measurement of the [[49, 1, 9]] code generators. Afterwards, error correction was performed blockwisely using the weight parity of the error in each sub-block, together with the syndrome obtained from the measurement of the seven-qubit code generators for such a sub-block. We also found some evidences suggesting that a similar error-correction technique may be applicable to other concatenated codes such as the concatenated Golay code and a concatenated Steane code with more than two levels of concatenation.

In this work, we use a different approach; we consider a case in which C_2 is not constructed from concatenating C_1's. In Sec. III, we consider the 3D color code of distance 3 in the form that has a 2D color code of distance 3 as a subcode, and we try to construct circuits for measuring its generators that give a distinguishable fault set. We extend the construction ideas to families of capped and recursive color codes in Sec. IV. Fault-tolerant protocols for the code and circuits that give a distinguishable fault set will be discussed in Sec. V.

III. SYNDROME MEASUREMENT CIRCUITS FOR THE 3D COLOR CODE OF DISTANCE 3

In this section, we try to find circuits for measuring generators of the 3D color code of distance 3 that gives a distinguishable fault set. We first define a 3D color code of distance 3 as a CSS subsystem code and observe some of its properties that are useful for fault-tolerant quantum computation. Afterwards, we give the CNOT orderings for the circuits that can make the fault set become distinguishable.

A. The 3D color code of distance 3

First, let us consider the qubit arrangement as displayed in Fig. 2(a). A 3D color code of distance 3 [73] is a [[15, 1, 3]] CSS subsystem code [76,77] that can be described by the stabilizer group $S_{3D} = \langle v_i^x, v_i^z\rangle$ and the gauge group $G_{3D} = \langle v_i^x, v_i^z, f_j^x, f_j^z\rangle$, $i = 0, 1, 2, 3$ and $j = 1, 2, \ldots, 6$, where the v_i^x and f_j^x (or the v_i^z and f_j^z) are X-type (or Z-type) operators defined on the following set of qubits:

(a) v_0^x (or v_0^z) is defined on $\mathtt{q}_0, \mathtt{q}_1, \mathtt{q}_2, \mathtt{q}_3, \mathtt{q}_4, \mathtt{q}_5, \mathtt{q}_6, \mathtt{q}_7$,
(b) v_1^x (or v_1^z) is defined on $\mathtt{q}_1, \mathtt{q}_2, \mathtt{q}_3, \mathtt{q}_5, \mathtt{q}_8, \mathtt{q}_9, \mathtt{q}_{10}, \mathtt{q}_{12}$,
(c) v_2^x (or v_2^z) is defined on $\mathtt{q}_1, \mathtt{q}_3, \mathtt{q}_4, \mathtt{q}_6, \mathtt{q}_8, \mathtt{q}_{10}, \mathtt{q}_{11}, \mathtt{q}_{13}$,
(d) v_3^x (or v_3^z) is defined on $\mathtt{q}_1, \mathtt{q}_2, \mathtt{q}_4, \mathtt{q}_7, \mathtt{q}_8, \mathtt{q}_9, \mathtt{q}_{11}, \mathtt{q}_{14}$,
(e) f_1^x (or f_4^z) is defined on $\mathtt{q}_1, \mathtt{q}_2, \mathtt{q}_3, \mathtt{q}_5$,
(f) f_2^x (or f_5^z) is defined on $\mathtt{q}_1, \mathtt{q}_3, \mathtt{q}_4, \mathtt{q}_6$,
(g) f_3^x (or f_6^z) is defined on $\mathtt{q}_1, \mathtt{q}_2, \mathtt{q}_4, \mathtt{q}_7$,
(h) f_4^x (or f_1^z) is defined on $\mathtt{q}_1, \mathtt{q}_4, \mathtt{q}_8, \mathtt{q}_{11}$,
(i) f_5^x (or f_2^z) is defined on $\mathtt{q}_1, \mathtt{q}_2, \mathtt{q}_8, \mathtt{q}_9$,
(j) f_6^x (or f_3^z) is defined on $\mathtt{q}_1, \mathtt{q}_3, \mathtt{q}_8, \mathtt{q}_{10}$.

Here qubit i in Fig. 2(a) is denoted by $\mathtt{q}_i$. Graphically, the v_i^x and v_i^z are the eight-body volumes shown in Fig. 2(b), and the f_j^x and f_j^z are the four-body faces shown in Fig. 2(c). Note that f_j^x and f_k^z anticommute when $j = k$, and they commute when $j \neq k$. The dual lattice of the 3D color code of distance 3 is illustrated in Fig. 2(d), where each vertex represents each stabilizer generator.

The 3D color code of distance 3 can be viewed as the [[15, 7, 3]] Hamming code in which six out of seven logical qubits become gauge qubits. From the subsystem code previously described, a [[15, 1, 3]] stabilizer code can be constructed by fixing some gauge qubits; i.e., choosing some gauge operators that commute with one another and including them in the stabilizer group. In this work, we discuss two possible ways to construct a stabilizer code from the 3D color code of distance 3. The resulting codes will be called the 3D color code in H form and the 3D color code in T form.

1. The 3D color code of distance 3 in H form

Let us consider the center plane of the code shown in Fig. 2(a) that covers $\mathtt{q}_1$ to $\mathtt{q}_7$. We can see that the plane looks exactly like the 2D color code of distance 3 [54], whose stabilizer group is $S_{2D} = \langle f_1^x, f_2^x, f_3^x, f_4^z, f_5^z, f_6^z\rangle$ (the 2D color code of distance 3 is equivalent to the [[7, 1, 3]]

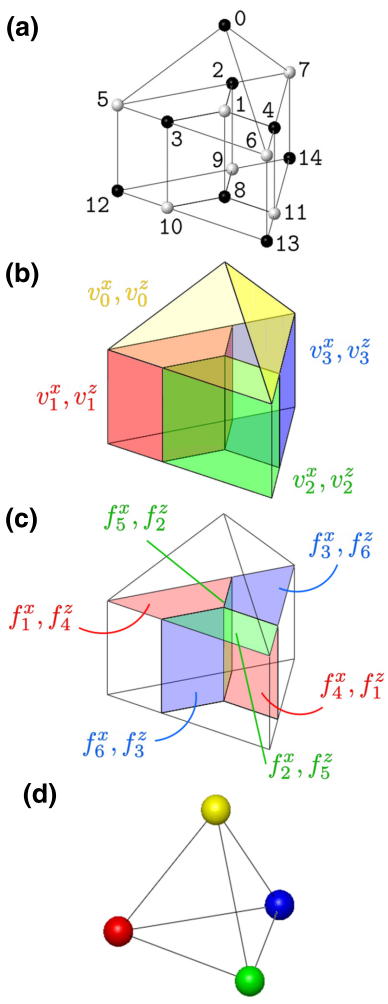


FIG. 2. The 3D color code of distance 3. In (a), qubits are represented by vertices. Note that the set of qubits are bipartite, as displayed by black and white colors. Stabilizer generators and gauge generators of the code are illustrated by volume operators in (b) and face operators in (c), respectively. The dual lattice of the code is shown in (d).

Steane code). The 3D color code in H form is constructed by adding the stabilizer generators of the 2D color code to the old generating set of the 3D color code; the stabilizer group of the 3D color code of distance 3 in H form is

$$S_H = \langle v_0^x, v_1^x, v_2^x, v_3^x, f_1^{\,x}, f_2^{\,x}, f_3^{\,x}, v_0^z, v_1^z, v_2^z, v_3^z, f_4^{\,z}, f_5^{\,z}, f_6^{\,z}\rangle. \tag{6}$$

We can choose logical X and logical Z operators of this code to be $X^{\otimes n}M$ and $Z^{\otimes n}N$ for some stabilizers $M, N \in S_H$. One important property of the code in H form for fault-tolerant quantum computation is that the logical Hadamard, S, and CNOT gates are transversal; i.e., $\bar{H} = H^{\otimes n}$ is a logical Hadamard gate, $\bar{S} = (S^\dagger)^{\otimes n}$ is a logical S gate, and $\overline{\text{CNOT}} = \text{CNOT}^{\otimes n}$ is a logical CNOT gate, where

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

Note that the choice of stabilizer generators for S_H is not unique. However, the choice of generators determines how the error syndrome will be measured, and different choices of generators can give different fault sets. The circuits for measuring generators discussed later in Sec. III B only correspond to the choice of generators in Eq. (6).

2. The 3D color code of distance 3 in T form

Compared to the code in H form, the 3D color code of distance 3 in T form is constructed from different gauge operators of the [[15, 1, 3]] subsystem code. In particular, the generators of the code in T form consist of the generators of the [[15, 1, 3]] subsystem code and all Z-type four-body face generators; i.e., the stabilizer group of the code in T form is

$$S_T = \langle v_0^x, v_1^x, v_2^x, v_3^x, f_1^{\,z}, f_2^{\,z}, f_3^{\,z}, v_0^z, v_1^z, v_2^z, v_3^z, f_4^{\,z}, f_5^{\,z}, f_6^{\,z}\rangle. \tag{7}$$

Similar to the code in H form, we can choose logical X and logical Z operators of this code to be $X^{\otimes n}M$ and $Z^{\otimes n}N$ for some stabilizers $M, N \in S_T$. Also, the CNOT gate is transversal in the code of T form. However, one major difference from the code in H form is that Hadamard and S gates are not transversal in this code. Instead, a T gate is transversal; a logical T gate can be implemented by applying T gates on all qubits represented by black vertices in Fig. 2(a) and applying $T^\dagger$ gates on all qubits represented by white vertices, where

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{i} \end{pmatrix}.$$

In fact, the code in T form is equivalent to the [[15, 1, 3]] quantum Reed-Muller code. Note that Lemma 1 is applicable to both codes in H form and T form since they have all code properties required by the lemma, even though X-type and Z-type generators are not similar in the case of the code in T form.

3. Code switching

It is possible to transform between the code in H form and the code in T form using the technique called *code switching* [71–74]. The process involves measurements of gauge operators of the [[15, 1, 3]] subsystem code, which can be done as follows. Suppose that we start from the code in H form. We can switch to the code in T form by first measuring $f_1^{\,z}, f_2^{\,z}$, and $f_3^{\,z}$. Afterwards, we must apply an X-type Pauli operator that

1. commutes with all the v_i^x and v_i^z $(i = 0, 1, 2, 3)$,
2. commutes with f_4^z, f_5^z, f_6^z, and
3. for each $j = 1, 2, 3$, commutes with f_j^z if the outcome from measuring such an operator is 0 (the eigenvalue is $+1$) or anticommutes with f_j^z if the outcome is 1 (the eigenvalue is -1).

Switching from the code in T form to the code in H form can be done similarly, except that f_1^x, f_2^x, and f_3^x will be measured and the operator to be applied must be a Z-type Pauli operator that commutes or anticommutes with f_1^x, f_2^x, and f_3^x (depending on the measurement outcomes).

Transversal gates satisfy the conditions for fault-tolerant gate gadgets proposed in Ref. [10] (see Sec. V A); thus, they are very useful for fault-tolerant quantum computation. It is known that universal quantum computation can be performed using only H, S, CNOT, and T gates [63,64,69]. However, for any QECC, universal quantum computation cannot be achieved using only transversal gates due to the Eastin-Knill theorem [60]. Fortunately, the code switching technique allows us to perform universal quantum computation using both codes in H form and T form; any logical Clifford gate can be performed transversally on the code in H form since the Clifford group can be generated by $\{H, S, \text{CNOT}\}$, and a logical T gate can be performed transversally on the code in T form. For the 3D color code of distance 3, code switching can be done fault-tolerantly using the above method [73,74] or the method presented in Ref. [75] that involves a logical Einstein-Podolsky-Rosen state.

B. Circuit configuration for the 3D color code of distance 3

In this section, circuits for measuring the generators of the 3D color code of distance 3 in H form will be developed. Here we try to find CNOT orderings for the circuits that make fault set $\mathcal{F}_1$ distinguishable (where $\mathcal{F}_1$ is the set of all fault combinations arising from up to one fault as defined in Definition 3). The ideas used for the circuit construction in this section will be later adapted to the circuits for measuring generators of a capped or a recursive color code (capped and recursive capped codes will be defined in Secs. IV A and IV B, and the circuit construction will be discussed in Sec. IV C). Fault-tolerant protocols for the 3D color code of distance 3 are similar to fault-tolerant protocols for capped color codes, which will be later discussed in Sec. V.

For simplicity, since X-type and Z-type data errors can be corrected separately and X-type and Z-type generators of our choice have the same form, we only discuss the case that a single fault can give rise to a Z-type data error. Similar analysis will also be applicable to the case of X-type errors. We start by observing that the 2D color code of distance 3 is a subcode of the 3D color code of distance 3 in H form, where the 2D color code lies on the center plane of the code illustrated in Fig. 2(a). The 2D color code is a code to which Lemma 1 is applicable, meaning that if we can measure the syndrome and the weight parity of any Z-type Pauli error that occurred on the center plane, we can always find a Pauli operator logically equivalent to such an error. Moreover, we can see that the generator v_0^x has support on all qubits on the center plane ($\mathtt{q}_1$ to $\mathtt{q}_7$). This means that the weight parity of a Z-type error on the center plane can be obtained by measuring v_0^x. For these reasons, we can always find an error-correction operator for any Z-type error that occurred on the center plane using the measurement outcomes of f_1^x, f_2^x, f_3^x (which give the syndrome of the error evaluated on the 2D color code) and the measurement outcome of v_0^x (which gives the weight parity of the error).

All circuits for measuring generators of the 3D color code in H form used in this section are nonflag circuits. Each circuit has w data CNOT gates, where w is the weight of the operator being measured. The circuit for each generator looks similar to the circuit in Fig. 3, but the ordering of data CNOT gates has yet to be determined.

Our goal is to find CNOT orderings for all circuits involved in the syndrome measurement so that $\mathcal{F}_1$ is distinguishable. Thus, we have to consider all possible errors arising from a single fault, not only the errors that occurred on the center plane. Let us first consider an arbitrary single fault that can lead to a purely Z-type error. Since the 3D color code in H form has distance 3, all Z-type errors of weight 1 correspond to different syndromes. All we have to worry about are single faults that can lead to a Z-type error of weight > 1 that has the same syndrome as some error of weight 1 but is not logically equivalent to such an error. Note that a Z-type error of weight > 1 arising from a single fault can only be caused by a faulty CNOT gate in some circuit for measuring a Z-type generator.

We can divide the generators of the 3D color code in H form into three categories:

1. `cap` generators, consisting of v_0^x and v_0^z,
2. `f` generators, consisting of $f_1^x, f_2^x, f_3^x, f_4^z, f_5^z, f_6^z$,
3. `v` generators, consisting of $v_1^x, v_2^x, v_3^x, v_1^z, v_2^z, v_3^z$.

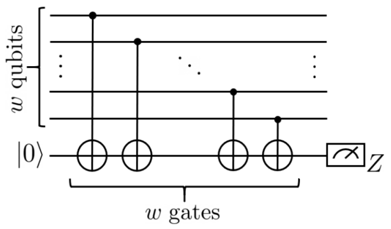


FIG. 3. A nonflag circuit for measuring a Z-type generator of weight w for the 3D color code of distance 3. The ordering of the CNOT gates for each generator has yet to be determined.

(We consider v_0^x and v_0^z separately from other $\mathtt{v}$ generators because they cover all qubits on the center plane.) Here we analyze the pattern of Z-type errors arising from the measurement of Z-type generators of each category. The syndrome of each Z-type error will be represented in the form $(u, \vec{v}, \vec{w})$, where $u, \vec{v}, \vec{w}$ are syndromes obtained from the measurement of $\mathtt{cap}$, $\mathtt{f}$, and $\mathtt{v}$ generators of X type, respectively. Note that, for each $\mathtt{v}$ generator, there will be only one $\mathtt{f}$ generator such that the set of supporting qubits of the $\mathtt{v}$ generator contains all supporting qubits of the $\mathtt{f}$ generator (for example, v_1^x and f_1^x, or v_1^z and f_4^z).

Let us start by observing the syndromes of any Z-type error of weight 1. An error on the following qubits gives the syndrome of the following form:

(a) an error on $\mathtt{q}_0$ gives syndrome $(1, \vec{0}, \vec{0})$,
(b) an error on $\mathtt{q}_i$ ($i = 1, \ldots, 7$) gives a syndrome of the form $(1, \vec{q}_i, \vec{q}_i)$,
(c) an error on $\mathtt{q}_{7+i}$ ($i = 1, \ldots, 7$) gives a syndrome of the form $(0, \vec{0}, \vec{q}_i)$.

Here $\vec{q}_i \in \mathbb{Z}_2^3$ is not zero (see Table I below as an example). We can see that all Z-type errors of weight 1 give different syndromes, as expected. Next, let us consider a Z-type error E of any weight that occurs only on the center plane. Suppose that the weight parity of E is WP (WP is 0 or 1), and the syndrome of E obtained from measuring f_1^x, f_2^x, f_3^x is $\vec{p}$. Then, the syndrome of E obtained from measuring all X-type generators is as follows:

(a) an error E on the center plane gives a syndrome of the form $(\mathrm{WP}, \vec{p}, \vec{p})$.

We find that the following statements hold.

1. Error E and the error on $\mathtt{q}_0$ will have the same syndrome if E has odd weight and $\vec{p}$ is trivial, which means that E is equivalent to $Z^{\otimes 7}$ on the center plane. In this case, E and Z_0 are logically equivalent up to a multiplication of v_0^z and some stabilizer.
2. Error E and an error on $\mathtt{q}_i$ ($i = 1, 2, \ldots, 7$) will have the same syndrome if E has odd weight and $\vec{p} = \vec{q}_i$ for some i. In this case, E and Z_i have the same weight parity and the same syndrome (evaluated by the generators of the 2D color code), meaning that E and Z_i are logically equivalent by Lemma 1.
3. Error E and an error on $\mathtt{q}_i$ ($i = 7, 8, \ldots, 14$) cannot have the same syndrome since $\vec{q}_i \neq \vec{0}$.

Therefore, a Z-type error of any weight that occurred only on the center plane either has a syndrome different from those of Z-type errors of weight 1, or is logically equivalent to some Z-type error of weight 1.

Because of the aforementioned properties of a Z-type error on the center plane, we try to design circuits for measuring Z-type generators so that most of the possible Z-type errors arising from a single fault are on the center plane. Finding a circuit for any $\mathtt{f}$ generator is easy since, for the 3D color code in H form, any $\mathtt{f}$ generator lies on the center plane, so any CNOT ordering will work. Finding a circuit for a $\mathtt{cap}$ generator is also easy; if the first data CNOT gate in the circuit is the one that couples $\mathtt{q}_0$ with the syndrome ancilla, we can make sure that all possible Z-type errors arising from a faulty CNOT gate in this circuit are on the center plane (up to a multiplication of v_0^z or v_0^x).

Finding a circuit for measuring a $\mathtt{v}$ generator is not obvious. Since some parts of any $\mathtt{v}$ generator of Z type are on the center plane and some parts are off the plane, some Z-type errors from a faulty data CNOT gate have support on some qubits that are not on the center plane. We want to make sure that in such cases, the error will not cause any problem; i.e., its syndrome must be different from those of other Z-type errors, or it must be logically equivalent to some Z-type error. In particular, we try to avoid the case that a CNOT fault can cause a Z error of weight > 1 that is totally off plane. This is because such a high-weight error and some Z_i with $i = 8, 9, \ldots, 14$ may have the same syndrome but they are not logically equivalent (for example, $Z_{10}Z_{12}$ and Z_{13} have the same syndrome but they are not logically equivalent).

One possible way to avoid such an error is to arrange the data CNOT gates so that the qubits on which they act are alternated between on-plane and off-plane qubits. An ordering of data CNOT gates used in the circuit for any $\mathtt{v}$ generator will be referenced by the ordering of data CNOT gates used in the circuit for its corresponding $\mathtt{f}$ generator. For example, if the ordering of data CNOT gates used for f_4^z is (2,5,3,1), then the ordering of data CNOT gates used for v_1^z will be (2,9,5,12,3,10,1,8). A configuration of data CNOT gates for a $\mathtt{v}$ generator similar to this setting will be called a *sawtooth configuration*. Using this configuration for every $\mathtt{v}$ generator, we find that there exists a CNOT ordering for each generator such that all possible (nonequivalent) Z-type errors from all circuits can be distinguished.

An example of the CNOT orderings that give a distinguishable fault set can be represented by the diagram in Fig. 4. The diagram looks similar to the 2D color code on the center plane, and thus all $\mathtt{f}$ generators are displayed. In the diagram

1. each arrow represents the ordering of data CNOT gates for each $\mathtt{f}$ generator: the qubits on which data CNOT gates act start from the qubit at the tail of an arrow, then proceed counterclockwise;
2. the ordering of data CNOT gates for each $\mathtt{v}$ generator can be obtained from its corresponding $\mathtt{f}$ generator using the sawtooth configuration;
3. the ordering of data CNOT gates for the $\mathtt{cap}$ generator is in numerical order.

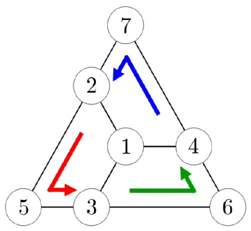


FIG. 4. An example of the orderings of CNOT gates for the 3D color code of distance 3 in H form which give a distinguishable fault set $\mathcal{F}_1$. For each `f` generator, the qubits on which data CNOT gates act start from the tail of each arrow, then proceed counterclockwise. The ordering of CNOT gates for the `cap` generator is determined by the qubit numbering.

From the diagram, the exact orderings of data CNOT gates for `f`, `v`, and `cap` generators are

1. (2,5,3,1), (3,6,4,1), and (4,7,2,1) for `f` generators;
2. (2,9,5,12,3,10,1,8), (3,10,6,13,4,11,1, 8), and (4,11, 7,14,2,9,1,8) for `v` generators;
3. (0,1,2,3,4,5,6,7) for the `cap` generator.

(Note that these are not the only CNOT orderings that give a distinguishable fault set.)

Possible Z-type errors of weight greater than 1 depend heavily on the ordering of CNOT gates in the circuits for measuring Z-type generators. The exhaustive list of all possible Z-type errors arising from one fault and their syndrome corresponding to the CNOT orderings in Fig. 4 is given in Table I. From the list, we find that any pair of possible Z-type errors either have different syndromes or are logically equivalent.

Since X-type and Z-type generators have the same form, this result is also applicable to the case of X-type errors. In general, a single fault in any circuit can cause an error of mixed type. However, note that a single fault in a circuit for measuring a Z-type generator cannot cause an X-type error of weight > 1 (and vice versa), and X-type and Z-type errors can be detected and corrected separately. Therefore, our results for X-type and Z-type errors imply that all fault combinations arising from up to one fault satisfy the condition in Definition 3. This means that $\mathcal{F}_1$ is distinguishable, and the protocols in Sec. V will be applicable. Since the circuits for measuring generators of the 3D color code are nonflag circuits, only one ancilla is required

TABLE I. All possible Z-type errors arising from one fault and their syndrome corresponding to the CNOT orderings in Fig. 4. Any pair of possible Z-type errors on the list either have different syndromes or are logically equivalent.

		Syndrome $(u, \vec{v}, \vec{w})$					Syndrome $(u, \vec{v}, \vec{w})$		
Fault origin	Error	u	$\vec{v}$	$\vec{w}$	Fault origin	Error	u	$\vec{v}$	$\vec{w}$
`q0`	Z_0	1	(0, 0, 0)	(0, 0, 0)	v_0^z	Z_0	1	(0, 0, 0)	(0, 0, 0)
`q1`	Z_1	1	(1, 1, 1)	(1, 1, 1)		Z_0Z_1	0	(1, 1, 1)	(1, 1, 1)
`q2`	Z_2	1	(1, 0, 1)	(1, 0, 1)		$Z_0Z_1Z_2$	1	(0, 1, 0)	(0, 1, 0)
`q3`	Z_3	1	(1, 1, 0)	(1, 1, 0)		$Z_0Z_1Z_2Z_3$	0	(1, 0, 0)	(1, 0, 0)
`q4`	Z_4	1	(0, 1, 1)	(0, 1, 1)		$Z_5Z_6Z_7$	1	(1, 1, 1)	(1, 1, 1)
`q5`	Z_5	1	(1, 0, 0)	(1, 0, 0)		Z_6Z_7	0	(0, 1, 1)	(0, 1, 1)
`q6`	Z_6	1	(0, 1, 0)	(0, 1, 0)		Z_7	1	(0, 0, 1)	(0, 0, 1)
`q7`	Z_7	1	(0, 0, 1)	(0, 0, 1)	v_1^z	Z_2	1	(1, 0, 1)	(1, 0, 1)
`q8`	Z_8	0	(0, 0, 0)	(1, 1, 1)		Z_2Z_9	1	(1, 0, 1)	(0, 0, 0)
`q9`	Z_9	0	(0, 0, 0)	(1, 0, 1)		$Z_2Z_9Z_5$	0	(0, 0, 1)	(1, 0, 0)
`q10`	Z_{10}	0	(0, 0, 0)	(1, 1, 0)		$Z_2Z_9Z_5Z_{12}$	0	(0, 0, 1)	(0, 0, 0)
`q11`	Z_{11}	0	(0, 0, 0)	(0, 1, 1)		$Z_{10}Z_1Z_8$	1	(1, 1, 1)	(1, 1, 0)
`q12`	Z_{12}	0	(0, 0, 0)	(1, 0, 0)		Z_1Z_8	1	(1, 1, 1)	(0, 0, 0)
`q13`	Z_{13}	0	(0, 0, 0)	(0, 1, 0)		Z_8	0	(0, 0, 0)	(1, 1, 1)
`q14`	Z_{14}	0	(0, 0, 0)	(0, 0, 1)	v_2^z	Z_3	1	(1, 1, 0)	(1, 1, 0)
f_4^z	Z_2	1	(1,0,1)	(1,0,1)		Z_3Z_{10}	1	(1, 1, 0)	(0, 0, 0)
	Z_2Z_5	0	(0,0,1)	(0,0,1)		$Z_3Z_{10}Z_6$	0	(1, 0, 0)	(0, 1, 0)
	Z_1	1	(1,1,1)	(1,1,1)		$Z_3Z_{10}Z_6Z_{13}$	0	(1, 0, 0)	(0, 0, 0)
f_5^z	Z_3	1	(1,1,0)	(1,1,0)		$Z_{11}Z_1Z_8$	1	(1, 1, 1)	(0, 1, 1)
	Z_3Z_6	0	(1,0,0)	(1,0,0)		Z_1Z_8	1	(1, 1, 1)	(0, 0, 0)
	Z_1	1	(1,1,1)	(1,1,1)		Z_8	0	(0, 0, 0)	(1, 1, 1)
f_6^z	Z_4	1	(0,1,1)	(0,1,1)	v_3^z	Z_4	1	(0, 1, 1)	(0, 1, 1)
	Z_4Z_7	0	(0,1,0)	(0,1,0)		Z_4Z_{11}	1	(0, 1, 1)	(0, 0, 0)
	Z_1	1	(1,1,1)	(1,1,1)		$Z_4Z_{11}Z_7$	0	(0, 1, 0)	(0, 0, 1)
						$Z_4Z_{11}Z_7Z_{14}$	0	(0, 1, 0)	(0, 0, 0)
						$Z_9Z_1Z_8$	1	(1, 1, 1)	(1, 0, 1)
						Z_1Z_8	1	(1, 1, 1)	(0, 0, 0)
						Z_8	0	(0, 0, 0)	(1, 1, 1)

in each protocol (assuming that the qubit preparation and measurement are fast and the ancilla can be reused).

In the next section, we generalize our technique to families of capped and recursive capped color codes, which have similar properties to the 3D color code of distance 3. Capped and recursive capped color codes will be defined in Secs. IV A and IV B respectively, and the construction of circuits for measuring the code generators will be discussed in Sec. IV C.

IV. SYNDROME MEASUREMENT CIRCUITS FOR A CAPPED COLOR CODE

In the previous section, we have seen that it is possible to construct circuits for the 3D color code of distance 3 such that the fault set is distinguishable. In this section, we extend our construction ideas to quantum codes of higher distance. First, we introduce families of capped and recursive capped color codes, whose properties are similar to those of the 3D color codes, but the structures of the recursive capped color codes of higher distance are more suitable for our construction rather than those of the 3D color codes of higher distance (as defined in Ref. [73]). Afterwards, we apply the error-correction ideas using weight parities from the previous section and develop the main theorem of this work, which can help us find proper CNOT orderings for a capped or a recursive capped color code of any distance.

A. Capped color codes

We begin by defining some notation for the 2D color codes [54] and stating some code properties. A 2D color code of distance d ($d = 3, 5, 7, \dots$) is an $[[n_{2D}, 1, d]]$ CSS-code, where $n_{2D} = (3d^2 + 1)/4$. The number of stabilizer generators of each type is $r = (n_{2D} - 1)/2$ (note that the total number of generators is $2r$). For any 2D color code, it is possible to choose generators so that those of each type (X or Z) are 3-colorable. The three smallest 2D color codes are shown in Fig. 5.

A 2D color code of any distance has the following properties [74]:

1. the number of qubits n_{2D} is odd,
2. every generator has even weight,
3. the code encodes one logical qubit,
4. logical X and logical Z operators are of the form $X^{\otimes n_{2D}}M$ and $Z^{\otimes n_{2D}}N$, where M, N are some stabilizers,
5. the set of physical qubits of a 2D color code is bipartite.

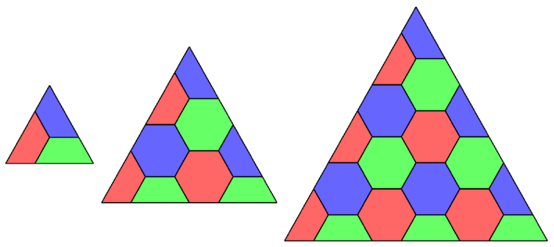


FIG. 5. 2D color codes of distances 3, 5, and 7.

With properties 1–4, we can see that Lemma 1 is applicable to a 2D color code of any distance.

A *capped color code* CCC(d) is constructed from two layers of the 2D color code of distance d plus one qubit. Thus, the number of qubits of CCC(d) is $2n_{2D} + 1 = 3(d^2 + 1)/2$. Examples of capped color codes with $d = 5$ and 7 are displayed in Fig. 6(a), and their dual lattices are shown in Fig. 6(b). Let $\mathfrak{q}_i$ denote qubit i. For convenience, we divide each code into three areas: the *top qubit* (consisting of $\mathfrak{q}_0$), the *center plane* (consisting of $\mathfrak{q}_1$ to $\mathfrak{q}_{n_{2D}}$), and the *bottom plane* (consisting of $\mathfrak{q}_{n_{2D}+1}$ to $\mathfrak{q}_{2n_{2D}}$). We primarily use the center plane as a reference, and sometimes call the qubits on the center plane *on-plane qubits* and call the qubits on the bottom plane *off-plane qubits*. Note that the set of physical qubits of CCC(d) is also bipartite [as colored in black and white in Fig. 6(a)] since the set of physical qubits of any 2D color code is bipartite.

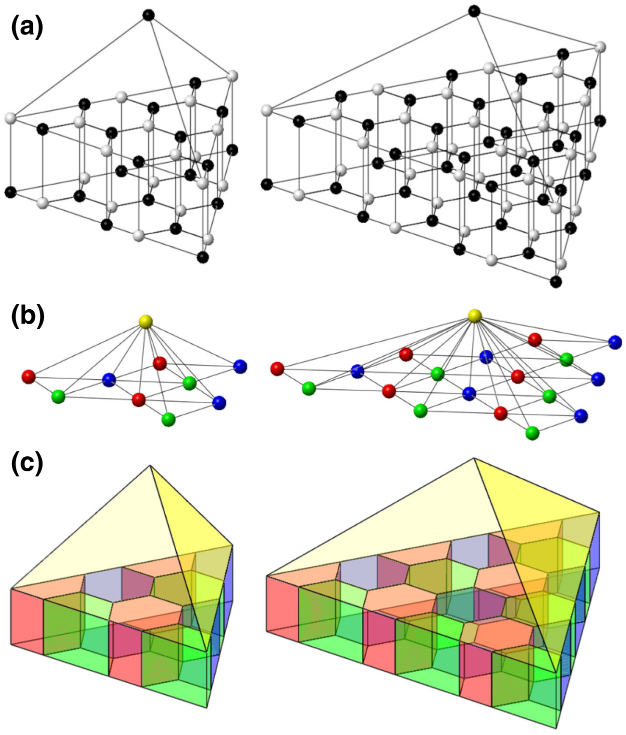


FIG. 6. Capped color codes CCC(d) with $d = 5$ (left) and $d = 7$ (right). (a) The set of qubits of any capped color code is bipartite, as displayed by black and white vertices. (b) The dual lattice of each capped color code. (c) Stabilizer generators of each code can be illustrated by volume operators.

A capped color code CCC(d) is a CSS subsystem code [76,77]. Its stabilizer generators are volume operators that can be defined as follows:

1. v_0^x and v_0^z are X-type and Z-type operators that cover q_0 and all qubits on the center plane—these operators are called cap generators,
2. $v_1^x, \dots, v_r^x$ and $v_1^z, \dots, v_r^z$ are X-type and Z-type operators in which each v_i^x (or v_i^z) acts as an X-type (or a Z-type) generator of the 2D color code on both center and bottom planes—these operators are called v generators.

The stabilizer generators of a capped color code are illustrated in Fig. 6(c). Using this notation, the stabilizer group of the code is

$$\mathcal{S}_{\text{CCC}} = \langle v_0^x, v_1^x, \dots, v_r^x, v_0^z, v_1^z, \dots, v_r^z \rangle. \tag{8}$$

For each CCC(d), the generators of the gauge group are face operators that can be defined as follows:

1. $f_1^x, \dots, f_r^x$ are X-type operators in which each operator acts as an X-type generator of the 2D color code on the center plane,
2. $f_{r+1}^z, \dots, f_{2r}^z$ are Z-type operators in which each operator acts as a Z-type generator of the 2D color code on the center plane, and f_i^x and f_{r+i}^z ($i = 1, \dots, r$) act on the same set of qubits,
3. $f_1^z, \dots, f_r^z$ and $f_{r+1}^x, \dots, f_{2r}^x$ are Z-type and X-type operators that satisfy the conditions
   (a) f_i^x and f_j^z anticommute when $i = j$ ($i,j = 1, \dots, 2r$),
   (b) f_i^x and f_j^z commute when $i \neq j$ ($i,j = 1, \dots, 2r$),
   (c) f_i^z and f_{r+i}^x ($i = 1, \dots, r$) act on the same set of qubits.

With this notation, the gauge group of each CCC(d) is

$$\mathcal{G}_{\text{CCC}} = \langle v_i^x, v_i^z, f_j^x, f_j^z \rangle, \tag{9}$$

where $i = 0, 1, \dots, r$ and $j = 1, \dots, 2r$.

Another way to define the gauge group of each CCC(d) is to use gauge generators of weight 4 that are vertical face operators lying between the center and the bottom planes, instead of $f_1^z, \dots, f_r^z$ and $f_{r+1}^x, \dots, f_{2r}^x$ defined previously. Let $e_1^z, \dots, e_r^z$ and $e_{r+1}^x, \dots, e_{2r}^x$ denote such generators (where e_i^z and e_{r+i}^x act on the same set of qubits). Each pair of e_i^z and e_{r+i}^x can be represented by an edge on a 2D color code. For example, vertical face generators $e_1^z, \dots, e_r^z$ and $e_{r+1}^x, \dots, e_{2r}^x$ of capped color codes with $d = 5$ and $d = 7$ are depicted in Fig. 7. Note that $\{f_1^z, \dots, f_r^z\}$ and $\{e_1^z, \dots, e_r^z\}$ (or $\{f_{r+1}^x, \dots, f_{2r}^x\}$ and $\{e_{r+1}^x, \dots, e_{2r}^x\}$) generate the same group. Therefore, the gauge group of each CCC(d) can also be written as

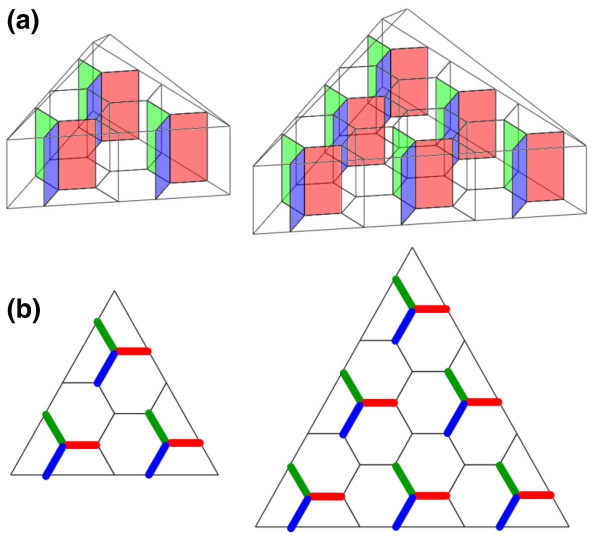


FIG. 7. (a) Vertical face generators $e_1^z, \dots, e_r^z$ and $e_{r+1}^x, \dots, e_{2r}^x$ of capped color codes CCC(d) with $d = 5$ (left) and $d = 7$ (right) (e_i^z and e_{r+i}^x act on the same set of qubits). The operators of each code can be represented by edges on a 2D color code, as shown in (b).

$$\mathcal{G}_{\text{CCC}} = \langle v_i^x, v_i^z, f_j^x, e_{r+j}^x, f_{r+j}^z, e_j^z \rangle, \tag{10}$$

where $i = 0, 1, \dots, r$ and $j = 1, \dots, r$.

It should be noted that in this work the term "color code" is used to describe a subsystem code satisfying the two conditions proposed in Ref. [74]. This may be different from common usages in other literature in which the term refers to a stabilizer code. A capped color code is actually a color code in three dimensions since the dual lattice of the code [see Fig. 6(b) for examples] is 4-colorable and can be constructed by attaching tetrahedra together (see Ref. [74] for more details). However, the capped color code and the 3D color code defined in Ref. [73] are different codes.

A capped color code is a subsystem code that encodes one logical qubit, meaning that there are n_{2D} gauge qubits for each CCC(d). We can clearly see that CCC(3) is exactly the 3D color code of distance 3 discussed in Sec. III A. Similarly, a stabilizer code encoding one logical qubit can be obtained from CCC(d) by choosing n_{2D} independent, commuting gauge operators, and including them in the stabilizer group. This work will discuss two possible ways to do so, and the resulting codes will be called the code in H form and the code in T form (similar to the case of the 3D color code of distance 3).

1. Capped color codes in H form

Observe that the center plane of CCC(d) that covers qubits 1 to n_{2D} looks exactly like the 2D color code of distance d. The stabilizer group of the 2D color code is

$S_{2D} = \langle f_1^x, \ldots, f_r^x, f_{r+1}^z, \ldots, f_{2r}^z \rangle$. A capped color code in H form constructed from CCC(d) can be obtained by adding the stabilizer generators of the 2D color code to the original generating set of CCC(d). Thus, the stabilizer group of the code in H form is

$$S_H = \langle v_i^x, v_i^z, f_j^x, f_{r+j}^z \rangle, \tag{11}$$

where $i = 0, 1, \ldots, r$ and $j = 1, 2, \ldots, r$. Logical X and logical Z operators of this code are of the form $X^{\otimes n}M$ and $Z^{\otimes n}N$, where M, N are some stabilizers in S_H. Note that Lemma 1 is applicable to the code in H form constructed from any CCC(d).

The code in H form is a code of distance d. This can be proved as follows.

Proposition 2. *The capped color code in H form constructed from* CCC(d) *has distance d.*

Proof. In order to prove that the distance of a stabilizer code is d, we show that the weight of a nontrivial logical operator is at least d; that is, any Pauli error of weight $< d$ is either a stabilizer or an error with a nontrivial syndrome, and there exists a nontrivial logical operator of weight exactly d. Since the capped color code in H form is a CSS code and X-type and Z-type generators have the same form, we can consider X-type and Z-type errors separately. For an error that occurred on the code in H form, we represent its weight by a triple (a, b, c), where a, b, c are the weights of the errors that occurred on the top qubit, the center plane, and the bottom plane, respectively.

Suppose that a Z-type error has weight $k < d$. The weight of such an error will be of the form (a, b, c) with $a = 0$ and $b + c = k$, or with $a = 1$ and $b + c = k - 1$. Observe that the stabilizer generators of the 2D color code on the center plane (which is a subcode of the capped color code in H form) are $f_1^x, \ldots, f_r^x$ and $f_{r+1}^z, \ldots, f_{2r}^z$. Moreover, the 2D color code on the bottom plane is also a subcode of the capped color code in H form, whose stabilizer generators are $f_1^x \cdot v_1^x, \ldots, f_r^x \cdot v_r^x$ and $f_{r+1}^z \cdot v_1^z, \ldots, f_{2r}^z \cdot v_r^z$ (the syndrome obtained by measuring v generators is the sum of the syndromes obtained from the 2D color codes on both planes). Since both 2D color codes on the center and the bottom planes have distance d, any Z-type error of weight $< d$ that occurs solely on the center or the bottom plane either has nontrivial syndrome or acts as a stabilizer on such a plane. From the possible forms of error, a Z-type error of weight $< d$ on the capped color code in H form corresponding to the trivial syndrome must act as a stabilizer on both planes and commute with v_0^x. Using Lemma 1, the weight of such an operator must be in the form $(0, b, c)$, where b, c are even numbers. So the total weight of the error is even, and it cannot be a logical Z operator (by Lemma 1). Therefore, any Z-type error of weight $< d$ is either a stabilizer or an error with a nontrivial syndrome. The same analysis is applicable to X-type errors of weight $< d$.

Next, we show that there exists a logical Z operator of weight exactly d. Consider a Z-type operator whose weight is of the form $(0, 0, d)$ and acts as a logical Z operator on the 2D color code on the bottom plane (the operator exists because the 2D color code has distance d). Such an operator commutes with all generators of the capped color code in H form and has odd weight. By Lemma 1, this operator is a logical Z operator. This completes the proof. ■

The capped color code in H form constructed from CCC(d) is an $[[n, 1, d]]$ code, where $n = 2n_{2D} + 1$. Similar to the 3D color code of distance 3 in H form, it is not hard to verify that Hadamard, S, and CNOT gates are transversal; their logical gates are $\bar{H} = H^{\otimes n}$, $\bar{S} = (S^\dagger)^{\otimes n}$, and $\overline{\text{CNOT}} = \text{CNOT}^{\otimes n}$.

It should be noted that there are many other choices of stabilizer generators that can give the same code as what is constructed here. However, different choices of generators can give different fault sets, which may or may not be distinguishable. In Sec. IV C, we only discuss circuits for measuring generators corresponding to Eq. (11).

2. Capped color codes in T form

A capped color code in T form is constructed from CCC(d) by adding all Z-type face generators of weight 4 to the old generating set of CCC(d). That is, the stabilizer group of the code in T form is

$$S_T = \langle v_i^x, v_i^z, f_j^z \rangle, \tag{12}$$

where $i = 0, 1, \ldots, r$ and $j = 1, 2, \ldots, 2r$, or, equivalently,

$$S_T = \langle v_i^x, v_i^z, e_k^z, f_{r+k}^z \rangle, \tag{13}$$

where $i = 0, 1, \ldots, r$ and $k = 1, 2, \ldots, r$. Similar to the code in H form, logical X and logical Z operators of this code are of the form $X^{\otimes n}M$ and $Z^{\otimes n}N$, where M, N are some stabilizers in S_T. Note that Lemma 1 is also applicable to the code in T form constructed for any CCC(d).

Unlike the code in H form, the capped color code in T form constructed from CCC(d) is a code of distance 3 regardless of the parameter d, i.e., it is an $[[n, 1, 3]]$ code, where $n = 2n_{2D} + 1$. The proof of the code distance is as follows.

Proposition 3. *The capped color code in T form constructed from* CCC(d) *has distance 3.*

Proof. Similar to the proof of Proposition 2, we show that (1) any Pauli error of weight < 3 is either a stabilizer or an error with a nontrivial syndrome, and that (2) there exists a nontrivial logical operator of weight exactly 3. However, for the capped color code in T form, X-type and Z-type generators have different forms, so we have to analyze both types of errors. Observe that all of the Z-type generators of the code in H form are also Z-type generators of the code in T form; thus, we can use the analysis in the proof of Proposition 2 to show that any X-type error of weight $< d$ is either a stabilizer or an error with a nontrivial syndrome. Thus, we only have to show that any Z-type error of weight < 3 is either a stabilizer or an error with a nontrivial syndrome, and that there exists a logical Z operator of weight exactly 3. Similar to the proof of Proposition 2, we represent its weight by a triple (a, b, c), where a, b, c are the weights of the errors that occurred on the top qubit, the center plane, and the bottom plane, respectively.

The X-type generators of the capped color code in T form are $v_0^x, v_1^x, \dots, v_r^x$. First, let us consider any Z-type error of weight 1. We can easily verify that the error anticommutes with at least one X-type generator, so its syndrome is nontrivial. Next, consider a Z-type error of weight 2. The weight of the error will have one of the following forms: $(0,2,0), (0,1,1), (0,0,2), (1,1,0)$, or $(1,0,1)$. We find that (1) a Z-type error of the form $(0,1,1)$ or $(1,0,1)$ anticommutes with v_0^x, and (2) a Z-type errors of the form $(0,2,0), (0,0,2)$, or $(1,1,0)$ anticommutes with at least one v generator (since v generators act as generators of the 2D color code on both planes simultaneously, and the 2D color code has distance d). Therefore, the syndrome of any Z-type error of weight 2 is nontrivial.

Next, we show that there exists a logical Z operator of distance exactly 3. Consider a Z-type operator of weight 3 of the form $Z_0 Z_i Z_{r+i}$, where $i = 1, 2, \dots, r$. We can verify that such an operator commutes with all X-type generators. Since the operator has odd weight, it is a logical Z operator by Lemma 1. ■

CNOT and T gates are transversal for the code in T form, while Hadamard and S gates are not. In order to prove the transversality of the T gate, we use the following lemma [74].

Lemma 2. *Let C be an $[[n,k,d]]$ CSS subsystem code in which n is odd, k is 1, and $X^{\otimes n}$ and $Z^{\otimes n}$ are bare logical X and Z operators [82]. Also, let Q be the set of all physical qubits of C, and let p be any positive integer. Suppose that there exists $V \subset Q$ such that, for any $m = 1, \dots, p$, for every subset $\{g_1^x, \dots, g_m^x\}$ of the X-type gauge generators of the code, the following holds:*

$$\left| V \cap \bigcap_{i=1}^{m} G_i \right| = \left| V^c \cap \bigcap_{i=1}^{m} G_i \right| \mod 2^{p-m+1} \tag{14}$$

where G_i is the set of physical qubits that support g_i^x. Then, a logical R_p gate (denoted by $\bar{R}_p$) can be implemented by applying R_p^q to all qubits in V and applying R_p^{-q} to all qubits in V^c, where $R_p = \mathrm{diag}(1, \exp(2\pi i/2^p))$, q is a solution to $q(|V| - |V^c|) = 1 \mod 2^p$, and $V^c = Q \backslash V$.

The proof of the transversality of the T gate is as follows.

Proposition 4. *A T gate is transversal for the capped color code in T form constructed from any* CCC(d).

Proof. Let C be the capped color code in T form constructed from any CCC(d) (C is a stabilizer code, i.e., it is a subsystem code in which the stabilizer group and the gauge group are the same). Note that the X-type stabilizer generators of the code are $v_0^x, v_1^x, \dots, v_r^x$, which are also the X-type gauge generators. Also, let $p = 3$ (since $T = R_3$), $q = 1$, and let V and V^c be the sets of qubits similar to those represented by the black and white vertices in Fig. 6(a) [this kind of representation is always possible for any CCC(d) since the set of physical qubits of CCC(d) is bipartite]. We use Lemma 2 and show that Eq. (14) is satisfied for $m = 1, 2, 3$.

Let G_i be the set of qubits that support X-type generator g_i^x. If $m = 1$, we can easily verify that $|V \cap G_1| = |V^c \cap G_1| \mod 8$ for every $g_1^x \in \{v_0^x, v_1^x, \dots, v_r^x\}$ since half of the supporting qubits of any X-type generator are in V and the other half are in V^c.

In the case when $m = 2$, let $\{g_1^x, g_2^x\}$ be a subset of $\{v_0^x, v_1^x, \dots, v_r^x\}$. If g_1^x is a cap generator v_0^x and g_2^x is a v generator $v_i^x, i = 1, \dots, r$, then $G_1 \cap G_2$ consists of the qubits that support the face generator f_i^x. Since half of the qubits in $G_1 \cap G_2$ are in V and the other half are in V^c, we have $|V \cap G_1 \cap G_2| = |V^c \cap G_1 \cap G_2|$ (equal to 2 or 3, depending on v_i^x). If g_1^x and g_2^x are adjacent v generators, then $G_1 \cap G_2$ consists of four qubits, two of them are in V and the other two are in V^c. So $|V \cap G_1 \cap G_2| = |V^c \cap G_1 \cap G_2| = 2$. If g_1^x and g_2^x are nonadjacent v generators, then $|V \cap G_1 \cap G_2| = |V^c \cap G_1 \cap G_2| = 0$. Therefore, Eq. (14) is satisfied for any subset $\{g_1^x, g_2^x\}$.

In the case when $m = 3$, let $\{g_1^x, g_2^x, g_3^x\}$ be a subset of $\{v_0^x, v_1^x, \dots, v_r^x\}$. If g_1^x is a cap generator v_0^x and g_2^x, g_3^x are adjacent v generators, or g_1^x, g_2^x, g_3^x are v generators in which any two of them are adjacent, then $G_1 \cap G_2 \cap G_3$ consists of two qubits, one of them is in V and the other one is in V^c. Thus, $|V \cap G_1 \cap G_2 \cap G_3| = |V^c \cap G_1 \cap G_2 \cap G_3| = 1$. If g_1^x is a cap generator v_0^x and g_2^x, g_3^x are nonadjacent v generators, or g_1^x, g_2^x, g_3^x are v generators in which some pair of them are not adjacent, then $G_1 \cap G_2 \cap G_3$ is the empty set. So $|V \cap G_1 \cap G_2 \cap G_3| = |V^c \cap G_1 \cap G_2 \cap G_3| = 0$. Therefore, Eq. (14) is satisfied for any subset $\{g_1^x, g_2^x, g_3^x\}$.

Since the sufficient condition in Lemma 2 is satisfied, a transversal T gate can be implemented by applying T gates to all qubits in V (represented by black vertices) and applying $T^\dagger$ gates to all qubits in V^c (represented by white vertices). ■

Incidentally, the capped color codes in T form presented here are similar to some codes that appear in other works. In fact, the capped color codes in T form are the same as the stacked codes with distance 3 protection defined in Ref. [78] (where alternative proofs of Propositions 3 and 4 are also presented). Such a code is the basis for the construction of the $(d-1)+1$ stacked code defined in the same work, whose code distance is d (see also Refs. [79,80] for other subsystem codes with similar construction).

3. Code switching

Similar to the 3D color code of distance 3, one can transform between the capped color code in H form and the code in T form derived from the same CCC(d) using the code switching technique [71–74]. Suppose that we start from the code in H form. The code switching can be done by first measuring $e_1^z, \dots, e_r^z$ (vertical face generators of weight 4), then applying an X-type Pauli operator that

1. commutes with all the v_i^x and v_i^z $(i = 0, 1, \dots, r)$,
2. commutes with $f_{r+1}^z, \dots, f_{2r}^z$, and
3. for each $j = 1, \dots, r$, commutes with e_j^z if the outcome from measuring such an operator is 0 (the eigenvalue is +1) or anticommutes with e_j^z if the outcome is 1 (the eigenvalue is -1).

We can use a similar process to switch from the code in T form to the code in H form, except that $f_1^x, \dots, f_r^x$ will be measured and the operator to be applied is a Z-type Pauli operator that commutes or anticommutes with $f_1^x, \dots, f_r^x$ (depending on the measurement outcomes).

B. Recursive capped color codes

One drawback of a capped color code CCC(d) is that the code in T form has only distance 3 regardless of the parameter d. This prevents us from performing fault-tolerant T-gate implementation through code switching because a few faults that occur to the code in T form can lead to a logical error. In this section, we introduce a way to construct a code of distance d from capped color codes through a process of recursive encoding. The resulting code will be called the *recursive capped color code*.

First, let us consider a capped color code in T form obtained from any (subsystem) capped color code CCC(d). There are many possible errors of weight 3 that are nontrivial logical errors of this code, but all of them have one thing in common: any logical error of weight 3 has support on $\mathfrak{q}_0$ (the top qubit of a capped color code). So if we can reduce the error rate on $\mathfrak{q}_0$, a logical error of weight 3 on a capped color code in T form will be less likely. In particular, if $\mathfrak{q}_0$ is encoded by a code of distance $d-2$, the distance of the resulting code will be d.

We define a *recursive capped color code* RCCC(d) ($d = 3, 5, 7, \dots$) to be a subsystem CSS code obtained from the following procedure:

1. RCCC(3) and CCC(3) are the same code,
2. RCCC(d) is obtained by encoding $\mathfrak{q}_0$ (the top qubit) of CCC(d) by RCCC($d-2$).

Constructing a recursive capped color code is similar to constructing a concatenated code. However, instead of encoding every physical qubit of the original code by another code, here we only encode $\mathfrak{q}_0$ of a capped color code by a recursive capped color code with smaller parameter.

It should be noted that a stacked code of distance d [78] can be obtained using a recursive encoding procedure similar to that presented above. However, in that case, the top qubit of a capped color code CCC(d) is encoded by the same capped color code [CCC(d)], and the procedure is repeated $(d-3)/2$ times. The recursive capped color code RCCC(d) with $d = 7$ and the stacked code of distance 7 are illustrated in Fig. 8.

The number of qubits of RCCC(d) is $(d^3 + 3d^2 + 3d - 3)/4$. For convenience, we divide each RCCC(d) into d layers.

1. The first layer consists of the top qubit $\mathfrak{q}_0$.
2. The $(j-1)$th layer where $j = 3, 5, \dots, d$ (which is similar to a 2D color code of distance j) will be called the center plane of inner CCC(j).
3. The jth layer where $j = 3, 5, \dots, d$ (which is similar to a 2D color code of distance j) will be called the bottom plane of inner CCC(j).

Similar to a capped color code, the stabilizer generators of RCCC(d) are defined by volume operators of X and Z types, and the gauge generators are defined by volume and face generators of X and Z types.

1. Recursive capped color codes in H form

The stabilizer group of a recursive capped color code in H form can be obtained by adding X- and Z-type face

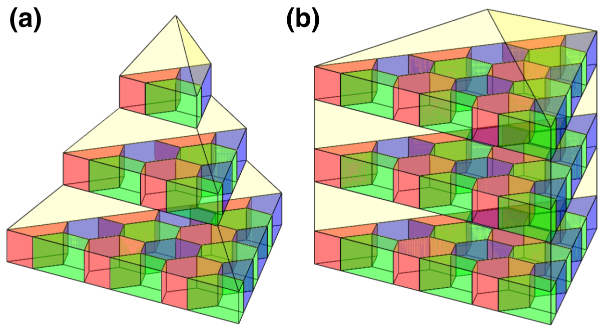


FIG. 8. (a) The recursive capped color code RCCC(d) with $d = 7$. (b) The stacked code of distance 7.

generators that are generators of 2D color codes on the center planes (layers $2, 4, \ldots, d-1$) to the original stabilizer generating set of RCCC(d). Similar to the construction of the subsystem code previously described, the recursive capped color code in H form constructed from RCCC(d) can also be obtained by encoding the top qubit $\mathfrak{q}_0$ of the capped color code in H form constructed from CCC(d) by the recursive capped color code in H form constructed from RCCC($d-2$).

The recursive capped color code in H form constructed from RCCC(d) has distance d. This can be proved as follows.

Proposition 5. *The recursive capped color code in H form constructed from* RCCC(d) *has distance d.*

Proof. Consider a capped color code in H form constructed from CCC(d) that has distance d (see Proposition 2). One example of a logical error of weight d of this code is a logical error of a 2D color code of distance d on the bottom plane. Encoding the top qubit of a capped color code in H form by the recursive capped color code in H form constructed from RCCC($d-2$) will not affect the aforementioned logical error, so the distance of the resulting code is still d. ■

A recursive capped color code in H form constructed from RCCC(d) is an $[[n, 1, d]]$ code, where $n = (d^3 + 3d^2 + 3d - 3)/4$. Similar to a capped color code in H form, a recursive capped color code in H form also possesses transversal Hadamard, S, and CNOT gates, where $\bar{H} = H^{\otimes n}$, $\overline{\text{CNOT}} = \text{CNOT}^{\otimes n}$, $\bar{S} = (S^\dagger)^{\otimes n}$ when $d = 3, 7, 11, \ldots$, and $\bar{S} = (S)^{\otimes n}$ when $d = 5, 9, 13, \ldots$.

2. Recursive capped color codes in T form

Consider the $(j-1)$th and jth layers ($j = 3, 5, \ldots, d$) of a subsystem code RCCC(d), which are similar to 2D color codes of distance j. We can define vertical face generators of inner CCC(j) between these two layers similar to the way we defined vertical face generators for CCC(d) in Sec. IV A (see Fig. 7 for examples). The stabilizer group of a recursive capped color code in T form can be obtained by adding vertical face generators of Z type of all inner CCC(j)'s to the original stabilizer generating set of RCCC(d). Also, similar to the construction of the subsystem code RCCC(d), the recursive capped color code in T form constructed from RCCC(d) can be obtained by encoding the top qubit $\mathfrak{q}_0$ of the capped color code in T form constructed from CCC(d) by the recursive capped color code in T form constructed from RCCC($d-2$).

Unlike the capped color code in T form constructed from CCC(d) whose distance is 3 regardless of the parameter d, the recursive capped color code in T form constructed from RCCC(d) has distance d. This can be proved as follows.

Proposition 6. *The recursive capped color code in T form constructed from* CCC(d) *has distance d.*

Proof. Consider a capped color code in T form constructed from CCC(d) that has distance 3 (see Proposition 3). We find that any logical error of weight 3 is of Z type and has support on $\mathfrak{q}_0$ (the top qubit of the capped color code). Suppose that $\mathfrak{q}_0$ is encoded by a code of distance $d-2$, effectively becoming an inner logical qubit $\bar{\mathfrak{q}}_0$. To create a logical error on the resulting code similar to the logical error of weight 3 on a capped color code in T form, we need an error on $\bar{\mathfrak{q}}_0$ plus errors on two more qubits. Thus, the minimum weight of a logical error of the resulting code is $(d-2)+2 = d$. In our case, the code being used to encode $\mathfrak{q}_0$ is the recursive capped color code in T form constructed from RCCC($d-2$). By induction, the recursive capped color code in T form constructed from RCCC(d) has distance d. ■

A recursive capped color code in T form constructed from RCCC(d) is an $[[n, 1, d]]$ code, where $n = (d^3 + 3d^2 + 3d - 3)/4$. Similar to a capped color code in T form, a recursive capped color code in T form also possesses transversal CNOT and T gates. The proof of transversality of the T gate is as follows.

Proposition 7. *A T gate is transversal for the recursive capped color code in T form constructed from any* RCCC(d).

Proof. Recall that, for any capped color code in T form, by Proposition 4, a logical T gate can be achieved by applying physical T and $T^\dagger$ gates on qubits represented by black and white vertices, respectively [see Fig. 6(a) for examples; the representation can be extended to any CCC(d) since the set of physical qubits of CCC(d) is bipartite]. Suppose that the top qubit $\mathfrak{q}_0$ of a capped color code in T form constructed from CCC(d) is encoded by the recursive capped color code in T form constructed from RCCC($d-2$), becoming an inner logical qubit $\bar{\mathfrak{q}}_0$. A logical T gate of the resulting code is similar to a logical T gate of the capped color code, except that an (inner) logical T gate is applied on $\bar{\mathfrak{q}}_0$. By induction, the T gate for $\bar{\mathfrak{q}}_0$ is transversal, and the T gate for the recursive capped color code in T form constructed from RCCC(d) is also transversal. ■

3. Code switching

Similar to the capped color codes, the code switching technique can be used to transform between the recursive capped color codes in H and T forms constructed from the same RCCC(d). In particular, we can switch from the code in H form to the code in T form by measuring Z-type vertical face generators of all inner CCC(j)'s and applying an appropriate Pauli operator depending on the measurement outcome. Switching from the code in T form to the code in H form can be done in a similar fashion, except that X-type generators of 2D color codes on the center planes (layers $2,4,\dots,d-1$) will be measured instead. We refer the reader to the process of finding a appropriate Pauli operator for code switching in Sec. IV A.

We have not yet discussed whether the procedure above is fault tolerant when we switch between the recursive capped color codes in H form and T form. However, the discussion of the fault-tolerant implementation of T gate will be deferred until Sec. V E.

C. Circuit configuration for capped and recursive capped color codes

One of the main goals of this work is to find circuits for measuring generators of a capped color code in H form in which the corresponding fault set $\mathcal{F}_t$ is distinguishable [where $t=\tau=(d-1)/2$ and $d=3,5,7,\dots$ is the code distance], and we expect that similar circuits will also work for a recursive capped color code in H form. As discussed before, the CNOT orderings and the number of flag ancillas are crucial for the circuit design. Finding such circuits for a capped color code of any distance using a random approach can be very challenging because of a few reasons. (1) The number of stabilizer generators of a capped color code increases quadratically as the distance increases. This means that the number of possible single faults in the circuits grow quadratically as well. (2) For a code with larger distance, a fault set $\mathcal{F}_t$ with larger t will be considered. Since it concerns all possible fault combinations arising from up to t faults, the size of $\mathcal{F}_t$ grows dramatically (perhaps exponentially) as t and the number of possible single faults increase. For these reasons, verifying whether $\mathcal{F}_t$ is distinguishable using the conditions in Definition 3 requires a lot of computational resources, and exhaustive searches for appropriate CNOT orderings may turn intractable.

Fortunately, there is a way to simplify the search for the CNOT orderings. From the structure of the capped color code in H form, it is possible to relate CNOT orderings for the 3D-like generators to those for the 2D-like generators, as we have seen in the circuit construction in Sec. III B. Instead of finding CNOT orderings directly for all generators, we simplify the problem and develop sufficient conditions for the CNOT orderings of the 2D-like generators that, if satisfied, can guarantee that the fault set $\mathcal{F}_t$ (which concerns both 3D-like and 2D-like generators) is distinguishable. Although we still need to check whether the sufficient conditions are satisfied for given CNOT orderings, the process is much simpler than checking the conditions in Definition 3 directly when the size of $\mathcal{F}_t$ is large.

We begin by dividing the stabilizer generators of the capped color code in H form constructed from CCC(d) into three categories (similar to the discussion in Sec. III B):

1. `cap` generators consisting of v_0^x and v_0^z,
2. `v` generators consisting of $v_1^x,\dots,v_r^x$ and $v_1^z,\dots,v_r^z$,
3. `f` generators consisting of $f_1^x,\dots,f_r^x$ and $f_{r+1}^z,\dots,f_{2r}^z$.

Here we only consider fault combinations arising from circuits for measuring Z-type generators that can lead to purely Z-type data errors of any weight. This is because i faults in circuits for measuring X-type generators cannot cause a Z-type data error of weight greater than i (and vice versa). Similar analysis will be applicable to the case of purely X-type errors, and also to the case of mixed-type errors. We first consider a Z-type data error and a flag vector arising from each single fault. Afterwards, fault combinations constructed from multiple faults will be considered, where the combined data error and the cumulative flag vector for each fault combination can be calculated using Eqs. (4) and (5).

Observe that the center plane of a capped color code behaves like a 2D color code, and that the weight of a Z-type error that occurred on the center plane can be measured by the `cap` generator v_0^x. In order to find CNOT orderings for generators of each category, we use an idea similar to that presented in Sec. III B; we try to design circuits for measuring Z-type generators so that most of the possible Z-type errors arising from a single fault are on the center plane. In this work, we start by imposing general configurations of data and flag CNOT gates; these general configurations will facilitate finding CNOT orderings. Then, exact configurations of CNOT gates that can make $\mathcal{F}_t$ distinguishable will be found using the theorem developed later in this section. The general configurations of data CNOT gates, which depend on the category of the generator, are as follows.

General configurations of data CNOT gates.

1. The `f` generator: there is no constraint for the ordering of data CNOT gates since each `f` generator lies on the center plane, but the ordering for f_{r+i}^z (or f_i^x) must be related to the ordering for v_i^z (or v_i^x) where $i=1,\dots,r$.
2. The `v` generator: the *sawtooth configuration* will be used; the qubits on which the data CNOT gates act must be alternated between on-plane and off-plane qubits. The ordering of data CNOT gates for v_i^z (or v_i^x) is referenced by the ordering of data CNOT gates for f_{r+i}^z (or f_i^x) where $i=1,\dots,r$ (see the examples in Fig. 9 and Sec. III B).
3. The `cap` generator: the first data CNOT gate must always be the one that couples $\mathtt{q}_0$ with the syndrome ancilla. The ordering of the other data CNOT gates has yet to be fixed.

In general, some flag ancillas may be added to the circuits for measuring a generator to help distinguish some possible errors and make $\mathcal{F}_t$ distinguishable. In that case, the general configurations for data CNOT gates will also be applied to the data CNOT gates in each flag circuit. Moreover, the following additional configurations for flag CNOT gates will be required.

General configurations of flag CNOT gates.

1. For each flag circuit, the first and the last data CNOT gates must not be in between any pair of flag CNOT gates.
2. The arrangements of flag CNOT gates in the circuits for each pair of $\mathtt{f}$ and $\mathtt{v}$ generators must be similar; suppose that a flag circuit for f_{r+i}^{z} (or f_i^{x}) where $i = 1, \ldots, r$ is given. A flag circuit for v_i^z (or v_i^x) is obtained by replacing each data CNOT gate that couples $\mathtt{q}_j$ with the syndrome ancilla ($j = 1, \ldots, n_{2D}$) by two data CNOT gates that couple $\mathtt{q}_j$ and $\mathtt{q}_{n_{2D}+j}$ with the syndrome ancilla; see the example in Fig. 9.

By imposing the general configurations for data and flag CNOT gates, what have yet to be determined before $\mathcal{F}_t$ is specified are the ordering of data CNOT gates for each $\mathtt{f}$ generator, the ordering of data CNOT gates after the first data CNOT gate for each $\mathtt{cap}$ generator, and the number of flag ancillas and the ordering of their relevant flag CNOT gates. (Note that having more flag ancillas can make fault distinguishing become easier, but more resources such as qubits and gates are also required.)

In this work, possible single faults that can give Z-type errors will be divided into seven types (based on relevant faulty locations) as follows.

1. Type $\mathtt{q}_0$: a fault causing a Z-type error on $\mathtt{q}_0$ that does not arise from any Z-type generator measurement. The total number of $\mathtt{q}_0$ faults is n_0 (which is 0 or 1).
2. Type $\mathtt{q}_{\mathtt{on}}$: a fault causing a single-qubit Z-type error on the center plane that does not arise from any Z-type generator measurement. The syndrome of an error is denoted by $\vec{q}_{\mathtt{on}}$. The total number of $\mathtt{q}_{\mathtt{on}}$ faults is $n_{\mathtt{on}}$.
3. Type $\mathtt{q}_{\mathtt{off}}$: a fault causing a single-qubit Z-type error on the bottom plane that does not arise from any Z-type generator measurement. The syndrome of an error is denoted by $\vec{q}_{\mathtt{off}}$. The total number of $\mathtt{q}_{\mathtt{off}}$ faults is $n_{\mathtt{off}}$.
4. Type $\mathtt{f}$: a fault that occurred during a measurement of a $\mathtt{f}$ generator of Z type. A Z-type error from each fault of this type and its syndrome are denoted by $\sigma_{\mathtt{f}}$ and $\vec{p}_{\mathtt{f}}$. A flag vector corresponding to each fault of this type is denoted by $\vec{f}_{\mathtt{f}}$. The total number of $\mathtt{f}$ faults is $n_{\mathtt{f}}$.
5. Type $\mathtt{v}$: a fault that occurred during a measurement of a $\mathtt{v}$ generator of Z type that gives errors of the same form on both center and bottom planes (see the example in Fig. 10). A part of a Z-type error from each fault of this type that occurred on the center plane only (or the bottom plane only) and its syndrome are denoted by $\sigma_{\mathtt{v}}$ and $\vec{p}_{\mathtt{v}}$. A flag vector corresponding to each fault of this type is denoted by $\vec{f}_{\mathtt{v}}$. The total number of $\mathtt{v}$ faults is $n_{\mathtt{v}}$.

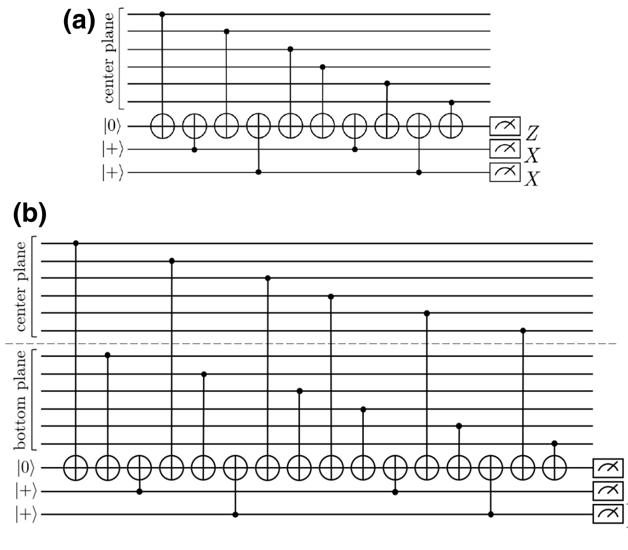


FIG. 9. (a) An example of a flag circuit for measuring the $\mathtt{f}$ generator with two flag ancillas. (b) A flag circuit for measuring the corresponding $\mathtt{v}$ generator. The circuit is obtained by replacing each data CNOT gate that couples $\mathtt{q}_j$ with the syndrome ancilla by two data CNOT gates that couple $\mathtt{q}_j$ and $\mathtt{q}_{n_{2D}+j}$ with the syndrome ancilla.

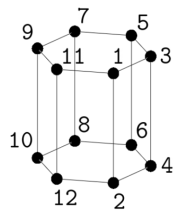


FIG. 10. Consider a circuit for measuring a $\mathtt{v}$ generator of Z type in which its supporting qubits are labeled as displayed above and the ordering of data CNOT gates is $(1, 2, \ldots, 12)$. A single fault in the circuit is either $\mathtt{v}$ type or $\mathtt{v}^*$ type, depending on whether the data errors on the center and the bottom planes have the same form. For example, an IZ fault on the seventh data CNOT gate is a $\mathtt{v}^*$ fault since the data error arising from the fault is $Z_9Z_{11} \otimes Z_8Z_{10}Z_{12}$, while an IZ fault on the eighth data CNOT gate is a $\mathtt{v}$ fault since the data error arising from the fault is $Z_9Z_{11} \otimes Z_{10}Z_{12}$.

6. Type $\mathtt{v}^*$: a fault that occurred during a measurement of a $\mathtt{v}$ generator of Z type in which an error that occurred on the center plane and an error on the bottom plane are different (see the example in Fig. 10). A part of a Z-type error from each fault of this type that occurred on the center plane only and its syndrome are denoted by $\sigma_{\mathtt{v}^*,\mathtt{cen}}$ and $\vec{p}_{\mathtt{v}^*,\mathtt{cen}}$. The other part of the Z-type error that occurred on the bottom plane only and its syndrome are denoted by $\sigma_{\mathtt{v}^*,\mathtt{bot}}$ and $\vec{p}_{\mathtt{v}^*,\mathtt{bot}}$. A flag vector corresponding to each fault of this type is denoted by $\vec{f}^*_{\mathtt{v}}$. The total number of $\mathtt{v}^*$ faults is $n^*_{\mathtt{v}}$.
7. Type $\mathtt{cap}$: a fault that occurred during a measurement of a $\mathtt{cap}$ generator of Z type. A Z-type error from each fault of this type and its syndrome are denoted by $\sigma_{\mathtt{cap}}$ and $\vec{p}_{\mathtt{cap}}$ ($\sigma_{\mathtt{cap}}$ is always on the center plane up to a multiplication of the $\mathtt{cap}$ generator being measured). A flag vector corresponding to each fault of this type is denoted by $\vec{f}_{\mathtt{cap}}$. The total number of $\mathtt{cap}$ faults is $n_{\mathtt{cap}}$.

Examples of faults of each type on the 3D structure are illustrated in Fig. 11(a). Note that a fault of $\mathtt{q}_0$, $\mathtt{q}_{\mathtt{on}}$, or $\mathtt{q}_{\mathtt{off}}$ type can be a Z-type input error, a single-qubit error from phase flip, or a single fault during any X-type generator measurement that gives a Z-type error.

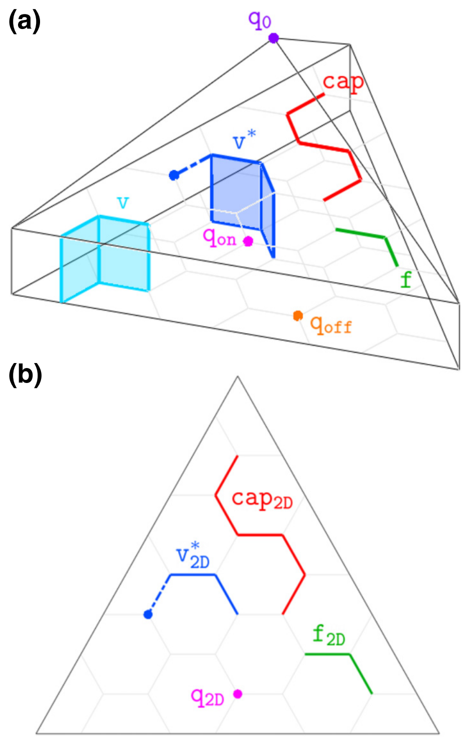


FIG. 11. (a) Examples of faults of each type on the 3D structure. (b) Examples of faults of each type on the 2D plane.

Suppose that a single fault causes a Z-type data error E and a flag vector $\vec{f}$. The syndrome of E evaluated by X-type generators can be written as $(s_a, \vec{s}_b, \vec{s}_c)$, where $s_a, \vec{s}_b, \vec{s}_c$ are syndromes obtained from measuring $\mathtt{cap}$, $\mathtt{f}$, and $\mathtt{v}$ generators of X type. In addition, the flag vector can be written as $(\vec{f}_a, \vec{f}_b, \vec{f}_c)$, where $\vec{f}_a, \vec{f}_b, \vec{f}_c$ are flag outcomes obtained from circuits for measuring $\mathtt{cap}$, $\mathtt{f}$, and $\mathtt{v}$ generators of Z type, respectively. (The lengths of $s_a, \vec{s}_b, \vec{s}_c$ are equal to the number of generators of each category, while the lengths of $\vec{f}_a, \vec{f}_b, \vec{f}_c$ are equal to the number of generators of each category times the number of flag ancillas in each flag circuit, assuming that all flag circuits have an equal number of flag ancillas.) Let $\mathrm{WP}(\sigma)$ denote the weight parity of error σ. Because of the general configurations of CNOT gates being used, the weight parity and the syndromes of a Z-type error (evaluated by X-type generators) and a flag vector arising from each type of fault can be summarized as in Table II. Note that, for a $\mathtt{v}^*$ fault, $\sigma_{\mathtt{v}^*,\mathtt{cen}}$ and $\sigma_{\mathtt{v}^*,\mathtt{bot}}$ differ by a Z error on a single qubit; i.e., $\mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{cen}}) + \mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{bot}}) = 1$. Sometimes we write $\vec{p}_{\mathtt{v}^*,\mathtt{cen}} + \vec{p}_{\mathtt{v}^*,\mathtt{bot}} = \vec{q}^*_{\mathtt{v}}$ to emphasize its similarity to the syndrome of a single-qubit error.

Now, let us consider the case that a fault combination arises from multiple faults. The syndrome and the weight parity of the combined error, and the cumulative flag vector of a fault combination can be calculated by adding the syndromes and the flag outcomes of all faults in the fault combination (the addition is modulo 2). For example, suppose that a fault combination consists of two faults that are of $\mathtt{q}_{\mathtt{on}}$ type and $\mathtt{v}$ type. The syndrome $\vec{s}(\mathbf{E})$ and the weight parity $\mathrm{WP}(\mathbf{E})$ of the combined error $\mathbf{E}$, and the cumulative flag vector $\vec{\mathbf{f}}$ that correspond to such a fault combination are

$$\vec{s}(\mathbf{E}) = (1 + \mathrm{WP}(\sigma_{\mathtt{v}}), \vec{q}_{\mathtt{on}} + \vec{p}_{\mathtt{v}}, \vec{q}_{\mathtt{on}}),$$
$$\mathrm{WP}(\mathbf{E}) = 1,$$
$$\vec{\mathbf{f}} = (\vec{0}, \vec{0}, \vec{f}_{\mathtt{v}}).$$

For a general fault combination composed of multiple faults, the corresponding syndrome, weight parity, and cumulative flag vector can be calculated as follows. Let $s_{\mathtt{cap}}, \vec{s}_{\mathtt{f}}, \vec{s}_{\mathtt{v}}$ denote syndromes of the combined error evaluated by $\mathtt{cap}$, $\mathtt{f}$, and $\mathtt{v}$ generators of X type, let $\mathrm{WP}_{\mathrm{tot}}$ denote the weight parity, and let $\vec{\mathbf{f}}_{\mathtt{cap}}, \vec{\mathbf{f}}_{\mathtt{f}}, \vec{\mathbf{f}}_{\mathtt{v}}$ denote parts of the cumulative flag vector obtained from circuits for measuring $\mathtt{cap}$, $\mathtt{f}$, and $\mathtt{v}$ generators of Z type. From Table II, we find that, for each fault combination,

$$s_{\mathtt{cap}} = n_0 + n_{\mathtt{on}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{cen}}) + \sum \mathrm{WP}(\sigma_{\mathtt{cap}}), \tag{15}$$

TABLE II. Syndrome $\vec{s} = (s_a, \vec{s}_b, \vec{s}_c)$, weight parity, and flag vector $\vec{f} = (\vec{f}_a, \vec{f}_b, \vec{f}_c)$ corresponding to a single fault of each type that leads to a Z-type error. Here $s_a, \vec{s}_b, \vec{s}_c$ are syndromes evaluated by `cap`, `f`, and `v` generators of X type, while $\vec{f}_a, \vec{f}_b, \vec{f}_c$ are flag outcomes obtained from circuits for measuring `cap`, `f`, and `v` generators of Z type. Note that in some cases a syndrome bit is equal to the weight parity of an error.

	Type of fault						
	$\mathtt{q}_0$	$\mathtt{q}_{\mathtt{on}}$	$\mathtt{q}_{\mathtt{off}}$	$\mathtt{f}$	$\mathtt{v}$	$\mathtt{v}^*$	$\mathtt{cap}$
s_a (cap)	1	1	0	$\mathrm{WP}(\sigma_{\mathtt{f}})$	$\mathrm{WP}(\sigma_{\mathtt{v}})$	$\mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{cen}})$	$\mathrm{WP}(\sigma_{\mathtt{cap}})$
$\vec{s}_b$ (f)	0	$\vec{q}_{\mathtt{on}}$	0	$\vec{p}_{\mathtt{f}}$	$\vec{p}_{\mathtt{v}}$	$\vec{p}_{\mathtt{v}^*,\mathtt{cen}}$	$\vec{p}_{\mathtt{cap}}$
$\vec{s}_c$ (v)	0	$\vec{q}_{\mathtt{on}}$	$\vec{q}_{\mathtt{off}}$	$\vec{p}_{\mathtt{f}}$	0	$\vec{p}_{\mathtt{v}^*,\mathtt{cen}} + \vec{p}_{\mathtt{v}^*,\mathtt{bot}}$ (or $\vec{q}^{\,*}_{\mathtt{v}}$)	$\vec{p}_{\mathtt{cap}}$
Weight parity	1	1	1	$\mathrm{WP}(\sigma_{\mathtt{f}})$	0	1	$\mathrm{WP}(\sigma_{\mathtt{cap}})$
$\vec{f}_a$ (cap)	0	0	0	0	0	0	$\vec{f}_{\mathtt{cap}}$
$\vec{f}_b$ (f)	0	0	0	$\vec{f}_{\mathtt{f}}$	0	0	0
$\vec{f}_c$ (v)	0	0	0	0	$\vec{f}_{\mathtt{v}}$	$\vec{f}^{\,*}_{\mathtt{v}}$	0

$$\vec{s}_{\mathtt{f}} = \sum \vec{q}_{\mathtt{on}} + \sum \vec{p}_{\mathtt{f}} + \sum \vec{p}_{\mathtt{v}} + \sum \vec{p}_{\mathtt{v}^*,\mathtt{cen}} + \sum \vec{p}_{\mathtt{cap}}, \tag{16}$$

$$\vec{s}_{\mathtt{v}} = \sum \vec{q}_{\mathtt{on}} + \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathtt{f}} + \sum \vec{q}^{\,*}_{\mathtt{v}} + \sum \vec{p}_{\mathtt{cap}}, \tag{17}$$

$$\mathrm{WP}_{\mathtt{tot}} = n_0 + n_{\mathtt{on}} + n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}}) + n^*_{\mathtt{v}} + \sum \mathrm{WP}(\sigma_{\mathtt{cap}}), \tag{18}$$

$$\vec{\mathbf{f}}_{\mathtt{cap}} = \sum \vec{f}_{\mathtt{cap}}, \tag{19}$$

$$\vec{\mathbf{f}}_{\mathtt{f}} = \sum \vec{f}_{\mathtt{f}}, \tag{20}$$

$$\vec{\mathbf{f}}_{\mathtt{v}} = \sum \vec{f}_{\mathtt{v}} + \sum \vec{f}^{\,*}_{\mathtt{v}}, \tag{21}$$

where each sum is over the same type of faults (the equations hold modulo 2). In addition, adding Eq. (15) to Eq. (18) and adding Eq. (16) to Eq. (17) we obtain

$$\mathrm{WP}_{\mathtt{bot}} = n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathtt{v}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{bot}}), \tag{22}$$

$$\vec{s}_{\mathtt{bot}} = \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathtt{v}} + \sum \vec{p}_{\mathtt{v}^*,\mathtt{bot}}, \tag{23}$$

where $\mathrm{WP}_{\mathtt{bot}} = s_{\mathtt{cap}} + \mathrm{WP}_{\mathtt{tot}}$ and $\vec{s}_{\mathtt{bot}} = \vec{s}_{\mathtt{f}} + \vec{s}_{\mathtt{v}}$.

Equations (15)–(23) are the main ingredients for the proof of the main theorem to be developed. One may notice that Eqs. (15) and (16), (17) and (18), and (22) and (23) come in pairs. They have the following physical meanings. Suppose that the combined error $\mathbf{E}$ is $\mathbf{E}_0 \cdot \mathbf{E}_{\mathtt{on}} \cdot \mathbf{E}_{\mathtt{off}}$, where $\mathbf{E}_0, \mathbf{E}_{\mathtt{on}}, \mathbf{E}_{\mathtt{off}}$ are the error on $\mathtt{q}_0$, the error on the center plane, and the error on the bottom plane. Then the following statements hold.

1. Equation (16) is the syndrome of $\mathbf{E}_{\mathtt{on}}$, while Eq. (15) is the weight parity $\mathbf{E}_{\mathtt{on}}$ plus the weight parity of $\mathbf{E}_0$.
2. Equation (17) is the syndrome of $\mathbf{E}_{\mathtt{on}} \cdot \mathbf{E}_{\mathtt{off}}$, while Eq. (18) is the weight parity of $\mathbf{E}_{\mathtt{on}} \cdot \mathbf{E}_{\mathtt{off}}$ plus the weight parity of $\mathbf{E}_0$. (Since `v` generators capture errors on both planes simultaneously, $\mathbf{E}_{\mathtt{on}} \cdot \mathbf{E}_{\mathtt{off}}$ can be viewed as a remaining error when $\mathbf{E}_{\mathtt{on}}$ and $\mathbf{E}_{\mathtt{off}}$ are "projected" on the same plane.)
3. Equation (23) is the syndrome of $\mathbf{E}_{\mathtt{off}}$, while Eq. (22) is the weight parity of $\mathbf{E}_{\mathtt{off}}$.

From these pairs of equations, and from the fact that now we only have to specify the ordering of data CNOT gates for each `f` generator, the ordering of data CNOT gates after the first gate for the `cap` generator, and the ordering of flag CNOT gates for each flag circuit, we can now simplify the CNOT ordering finding problem for a 3D structure to the problem of finding CNOT orderings on a 2D plane (which is similar to the 2D color code of distance d). In particular, each pair of equations concerns errors on a 2D plane (the center, the bottom, or the projected plane). We try to find conditions for the CNOT orderings on a 2D plane such that if satisfied, a bad case that makes $\mathcal{F}_t$ indistinguishable cannot happen.

Some types of faults on the 3D structure can be considered as the same types of faults when the problem is simplified. In the following we present types of possible single faults on the 2D plane and their correspondence on the 3D structure.

1. Type $\mathtt{q}_{\mathtt{2D}}$: a fault causing a single-qubit Z-type error on the 2D plane that does not arise from any Z-type generator measurement. The syndrome of an error is denoted by $\vec{q}_{\mathtt{2D}}$. The total number of $\mathtt{q}_{\mathtt{2D}}$ faults is $n_{\mathtt{q}_{\mathtt{2D}}}$. The combined error from only $\mathtt{q}_{\mathtt{2D}}$ faults is denoted by $\mathbf{E}_{\mathtt{q}_{\mathtt{2D}}}$. This type of

fault corresponds to $\mathtt{q}_{\mathtt{on}}$ and $\mathtt{q}_{\mathtt{off}}$ faults on the 3D structure.

2. Type $\mathtt{f}_{2\mathrm{D}}$: a fault that occurred during a measurement of a $\mathtt{f}$ generator of Z type. A Z-type error from each fault of this type and its syndrome are denoted by $\sigma_{\mathtt{f}_{2\mathrm{D}}}$ and $\vec{p}_{\mathtt{f}_{2\mathrm{D}}}$. A flag vector corresponding to each fault of this type is denoted by $\vec{f}_{\mathtt{f}_{2\mathrm{D}}}$. The total number of $\mathtt{f}_{2\mathrm{D}}$ faults is $n_{\mathtt{f}_{2\mathrm{D}}}$. The combined error from only $\mathtt{f}_{2\mathrm{D}}$ faults is denoted by $\mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$. This type of fault corresponds to $\mathtt{f}$ and $\mathtt{v}$ faults on the 3D structure (since an error on the center plane and an error on the bottom plane from a $\mathtt{v}$ fault have the same form; see the example in Fig. 10).
3. Type $\mathtt{v}^*_{2\mathrm{D}}$: a fault that occurred during a measurement of a $\mathtt{v}$ generator of Z type in which an error that occurred on the center plane and an error that occurred on the bottom plane are different (see the example in Fig. 10). A part of a Z-type error from each fault of this type that occurred on the center plane only and its syndrome are denoted by $\sigma_{\mathtt{v}^*_{2\mathrm{D}},\mathtt{cen}}$ and $\vec{p}_{\mathtt{v}^*_{2\mathrm{D}},\mathtt{cen}}$. The other part of the Z-type error that occurred on the bottom plane only and its syndrome are denoted by $\sigma_{\mathtt{v}^*_{2\mathrm{D}},\mathtt{bot}}$ and $\vec{p}_{\mathtt{v}^*_{2\mathrm{D}},\mathtt{bot}}$. A flag vector corresponding to each fault of this type is denoted by $\vec{f}_{\mathtt{v}^*_{2\mathrm{D}}}$. The total number of $\mathtt{v}^*_{2\mathrm{D}}$ faults is $n_{\mathtt{v}^*_{2\mathrm{D}}}$. The part of the combined error from only $\mathtt{v}^*_{2\mathrm{D}}$ faults on the center plane and the part on the bottom plane are denoted by $\mathbf{E}_{\mathtt{v}^*_{2\mathrm{D}},\mathtt{cen}}$ and $\mathbf{E}_{\mathtt{v}^*_{2\mathrm{D}},\mathtt{bot}}$. This type of fault corresponds to $\mathtt{v}^*$ faults on the 3D structure. (Note that this is the only type of fault that cannot be represented completely on the 2D plane since the error on the center plane and the error on the bottom plane are different. However, when running a computer simulation, we can treat a fault of $\mathtt{v}^*_{2\mathrm{D}}$ type similarly to a fault of $\mathtt{f}_{2\mathrm{D}}$ type except that two values of errors will be assigned to each fault.)
4. Type $\mathtt{cap}_{2\mathrm{D}}$: a fault that occurred during a measurement of a $\mathtt{cap}$ generator of Z type. A Z-type error from each fault of this type and its syndrome are denoted by $\sigma_{\mathtt{cap}_{2\mathrm{D}}}$ and $\vec{p}_{\mathtt{cap}_{2\mathrm{D}}}$ ($\sigma_{\mathtt{cap}_{2\mathrm{D}}}$ is always on the center plane up to a multiplication of the $\mathtt{cap}$ generator being measured). A flag vector corresponding to each fault of this type is denoted by $\vec{f}_{\mathtt{cap}_{2\mathrm{D}}}$. The total number of $\mathtt{cap}_{2\mathrm{D}}$ faults is $n_{\mathtt{cap}_{2\mathrm{D}}}$. The combined error from only $\mathtt{cap}_{2\mathrm{D}}$ faults is denoted by $\mathbf{E}_{\mathtt{cap}_{2\mathrm{D}}}$. This type of fault corresponds to $\mathtt{cap}$ faults on the 3D structure.

Examples of faults of each type on the 2D plane are illustrated in Fig. 11(b). The correspondence between the notation for types of faults on the 2D plane and the 3D structure is summarized in Table III.

We can see that possible Z-type errors on the 2D plane depend on the CNOT orderings for measuring $\mathtt{f}$ and $\mathtt{cap}$ generators of Z type. Next, we state the sufficient conditions for the CNOT orderings on the 2D plane that will make $\mathcal{F}_t$ (which concerns fault combinations from the 3D structure) distinguishable. These sufficient conditions are introduced in order to prevent the case that can lead to an "indistinguishable" pair (a pair of fault combinations from the 3D structure that does not satisfy any condition in Definition 3).

First, we state a condition that is automatically satisfied if a code being considered on the 2D plane is a code of distance d to which Lemma 1 is applicable.

Condition 0. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{q}_{2\mathrm{D}}} \leq d-1$, $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}}$ is not a nontrivial logical operator; equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{q}_{2\mathrm{D}} \neq 0 \mod 2$,
2. $n_{\mathtt{q}_{2\mathrm{D}}} \neq 1 \mod 2$.

Note that a nontrivial logical operator is an error corresponding to the trivial syndrome whose weight parity is odd (from Lemma 1). Condition 0 is equivalent to the fact that an error of weight $\leq d-1$ is detectable by a code of distance d; i.e., it either has a nontrivial syndrome or is a stabilizer. We state Condition 0 explicitly (although it is automatically satisfied) because the condition in this form looks similar to other conditions, which will simplify the proof of the main theorem.

We now state five sufficient conditions for the CNOT orderings on the 2D plane that will make $\mathcal{F}_t$ distinguishable.

Condition 1. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{f}_{2\mathrm{D}}} \leq d-2$, $\mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is not a nontrivial logical operator or the cumulative flag vector is not zero; equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{p}_{\mathtt{f}_{2\mathrm{D}}} \neq 0 \mod 2$,
2. $\sum \mathrm{WP}(\sigma_{\mathtt{f}_{2\mathrm{D}}}) \neq 1 \mod 2$,
3. $\sum \vec{f}_{\mathtt{f}_{2\mathrm{D}}} \neq 0 \mod 2$.

Condition 2. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{q}_{2\mathrm{D}}} + n_{\mathtt{f}_{2\mathrm{D}}} \leq d-3$, $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is not a nontrivial logical operator or the cumulative flag vector is not zero; equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{q}_{2\mathrm{D}} + \sum \vec{p}_{\mathtt{f}_{2\mathrm{D}}} \neq 0 \mod 2$,
2. $n_{\mathtt{q}_{2\mathrm{D}}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}_{2\mathrm{D}}}) \neq 1 \mod 2$,
3. $\sum \vec{f}_{\mathtt{f}_{2\mathrm{D}}} \neq 0 \mod 2$.

TABLE III. The correspondence between the notation for types of faults on the 2D plane and the 3D structure.

2D plane				3D structure			
Fault type	Syndrome	Weight parity	Flag vector	Fault type	Syndrome	Weight parity	Flag vector
$\mathtt{q}_{\mathtt{2D}}$	$\vec{q}_{\mathtt{2D}}$	1	$\vec{0}$	$\mathtt{q}_{\mathtt{on}}, \mathtt{q}_{\mathtt{off}}$, or $\mathtt{q}_{\mathtt{v}^*}$	$\vec{q}_{\mathtt{on}}, \vec{q}_{\mathtt{off}}$, or $\vec{q}^*_{\mathtt{v}}$	1	$\vec{0}$
$\mathtt{f}_{\mathtt{2D}}$	$\vec{p}_{\mathtt{f}_{\mathtt{2D}}}$	$\mathrm{WP}(\sigma_{\mathtt{f}_{\mathtt{2D}}})$	$\vec{f}_{\mathtt{f}_{\mathtt{2D}}}$	$\mathtt{f}, \mathtt{v}$, or $\mathtt{v}^*$	$\vec{p}_{\mathtt{f}}, \vec{p}_{\mathtt{v}}$, $\vec{p}_{\mathtt{v}^*,\mathtt{cen}}$, or $\vec{p}_{\mathtt{v}^*,\mathtt{bot}}$	$\mathrm{WP}(\sigma_{\mathtt{f}})$, $\mathrm{WP}(\sigma_{\mathtt{v}})$, $\mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{cen}})$, or $\mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{bot}})$	$\vec{f}_{\mathtt{f}}, \vec{f}_{\mathtt{v}}$, or $\vec{f}^*_{\mathtt{v}}$
$\mathtt{v}^*_{\mathtt{2D}}$	$\vec{p}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}}$	$\mathrm{WP}(\sigma_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}})$	$\vec{f}_{\mathtt{v}^*_{\mathtt{2D}}}$	$\mathtt{v}^*$	$\vec{p}_{\mathtt{v}^*,\mathtt{cen}}$	$\mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{cen}})$	$\vec{f}^*_{\mathtt{v}}$
	$\vec{p}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}}$	$\mathrm{WP}(\sigma_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}})$			$\vec{p}_{\mathtt{v}^*,\mathtt{bot}}$	$\mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{bot}})$	
$\mathtt{cap}_{\mathtt{2D}}$	$\vec{p}_{\mathtt{cap}_{\mathtt{2D}}}$	$\mathrm{WP}(\sigma_{\mathtt{cap}_{\mathtt{2D}}})$	$\vec{f}_{\mathtt{cap}_{\mathtt{2D}}}$	$\mathtt{cap}$	$\vec{p}_{\mathtt{cap}}$	$\mathrm{WP}(\sigma_{\mathtt{cap}})$	$\vec{f}_{\mathtt{cap}}$

Condition 3. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{f}_{\mathtt{2D}}} = 1$ and $n_{\mathtt{q}_{\mathtt{2D}}} + n_{\mathtt{f}_{\mathtt{2D}}} \leq d-2$, $\mathbf{E}_{\mathtt{q}_{\mathtt{2D}}} \cdot \mathbf{E}_{\mathtt{f}_{\mathtt{2D}}}$ is not a nontrivial logical operator or the cumulative flag vector is not zero; equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{q}_{\mathtt{2D}} + \sum \vec{p}_{\mathtt{f}_{\mathtt{2D}}} \neq 0 \mod 2$,
2. $n_{\mathtt{q}_{\mathtt{2D}}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}_{\mathtt{2D}}}) \neq 1 \mod 2$,
3. $\sum \vec{f}_{\mathtt{f}_{\mathtt{2D}}} \neq 0 \mod 2$.

Condition 4. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{f}_{\mathtt{2D}}} = 1$, $n_{\mathtt{q}_{\mathtt{2D}}} \geq 1$, $n_{\mathtt{v}^*_{\mathtt{2D}}} \geq 2$, and $n_{\mathtt{q}_{\mathtt{2D}}} + n_{\mathtt{f}_{\mathtt{2D}}} + n_{\mathtt{v}^*_{\mathtt{2D}}} = d-1$, the following does not happen: $\mathbf{E}_{\mathtt{f}_{\mathtt{2D}}} \cdot \mathbf{E}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}}$ is a stabilizer, $\mathbf{E}_{\mathtt{q}_{\mathtt{2D}}} \cdot \mathbf{E}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}}$ is a nontrivial logical operator, and the cumulative flag vector is zero. Equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{p}_{\mathtt{f}_{\mathtt{2D}}} + \sum \vec{p}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}} \neq 0 \mod 2$,
2. $\sum \mathrm{WP}(\sigma_{\mathtt{f}_{\mathtt{2D}}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}}) \neq 0 \mod 2$,
3. $\sum \vec{q}_{\mathtt{2D}} + \sum \vec{p}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}} \neq 0 \mod 2$,
4. $n_{\mathtt{q}_{\mathtt{2D}}} + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}}) \neq 1 \mod 2$,
5. $\sum \vec{f}_{\mathtt{f}_{\mathtt{2D}}} \neq 0 \mod 2$,
6. $\sum \vec{f}_{\mathtt{v}^*_{\mathtt{2D}}} \neq 0 \mod 2$.

Condition 5. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{cap}_{\mathtt{2D}}} = 1$, $n_{\mathtt{q}_{\mathtt{2D}}} \geq 1$, $n_{\mathtt{f}_{\mathtt{2D}}} + n_{\mathtt{v}^*_{\mathtt{2D}}} \geq 2$, and $n_{\mathtt{q}_{\mathtt{2D}}} + n_{\mathtt{f}_{\mathtt{2D}}} + n_{\mathtt{v}^*_{\mathtt{2D}}} + n_{\mathtt{cap}_{\mathtt{2D}}} = d-1$, the following does not happen: $\mathbf{E}_{\mathtt{f}_{\mathtt{2D}}} \cdot \mathbf{E}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}} \cdot \mathbf{E}_{\mathtt{cap}_{\mathtt{2D}}}$ is a stabilizer, $\mathbf{E}_{\mathtt{q}_{\mathtt{2D}}} \cdot \mathbf{E}_{\mathtt{f}_{\mathtt{2D}}} \cdot \mathbf{E}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}}$ is a nontrivial logical operator, and the cumulative flag vector is zero. Equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{p}_{\mathtt{f}_{\mathtt{2D}}} + \sum \vec{p}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}} + \sum \vec{p}_{\mathtt{cap}_{\mathtt{2D}}} \neq 0 \mod 2$,
2. $\sum \mathrm{WP}(\sigma_{\mathtt{f}_{\mathtt{2D}}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{cen}}) + \sum \mathrm{WP}(\sigma_{\mathtt{cap}_{\mathtt{2D}}}) \neq 0 \mod 2$,
3. $\sum \vec{q}_{\mathtt{2D}} + \sum \vec{p}_{\mathtt{f}_{\mathtt{2D}}} + \sum \vec{p}_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}} \neq 0 \mod 2$,
4. $n_{\mathtt{q}_{\mathtt{2D}}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}_{\mathtt{2D}}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*_{\mathtt{2D}},\mathtt{bot}}) \neq 1 \mod 2$,
5. $\sum \vec{f}_{\mathtt{f}_{\mathtt{2D}}} + \vec{f}_{\mathtt{v}^*_{\mathtt{2D}}} \neq 0 \mod 2$,
6. $\sum \vec{f}_{\mathtt{cap}_{\mathtt{2D}}} \neq 0 \mod 2$.

Conditions 1–5 prevent fault combinations of some form from occurring on the 2D plane (such fault combinations can lead to an indistinguishable fault set). If we arrange the CNOT gates in the circuits for f and cap generators so that all conditions are satisfied, then a fault set $\mathcal{F}_t$ (which considers the 3D structure) will be distinguishable. The main theorem of this work is as follows.

Theorem 1. *Let $\mathcal{F}_t$ be the fault set corresponding to circuits for measuring* f, v, *and* cap *generators of the capped color code in H form constructed from* CCC(d) *[where $t = (d-1)/2$, $d = 3, 5, 7, \ldots$], and suppose that the general configurations of* CNOT *gates for* f, v, *and* cap *generators are imposed, and that the circuits for each pair of X-type and Z-type generators use the same* CNOT *ordering. Let the code on the (simplified) 2D plane be the 2D color code of distance d. If all possible fault combinations on the 2D plane arising from the circuits for measuring* f *and* cap *generators satisfy Conditions 1 to 5, then $\mathcal{F}_t$ is distinguishable.*

Proof ideas. Theorem 1 is proved in Appendix A. The proof is organized as follows. First, we try to show that if Conditions 1–5 are satisfied, then for any fault combination arising from up to $d-1$ faults whose combined error is purely Z type, the fault combination cannot lead to a logical Z operator and the zero cumulative flag vector. The same analysis is also applicable to fault combinations whose combined error is purely X type since the circuits for measuring each pair of X-type and Z-type generators are of the same form. Afterwards, we use the fact that i

faults during the measurements of Z-type generators cannot cause an X-type error of weight more than i (and vice versa), and show that there is no fault combination arising from up to $d-1$ faults that leads to a nontrivial logical operator and the zero cumulative flag vector. By Proposition 1, this implies that $\mathcal{F}_t$ is distinguishable.

In order to prove the first part, we assume that Conditions 1–5 are satisfied and that there exists a fault combination arising from $< d$ faults whose combined error is a logical Z operator and its cumulative flag vector is zero; then we show that some contradiction will happen. From Lemma 1, a logical Z operator is a Z-type error with trivial syndrome and odd weight parity. Therefore, such a fault combination will give $s_{\texttt{cap}}=0$, $\vec{s}_{\texttt{f}}=\vec{0}$, $\vec{s}_{\texttt{v}}=\vec{0}$, $\mathrm{WP}_{\texttt{tot}}=1$, $\vec{\mathbf{f}}_{\texttt{cap}}=\vec{0}$, $\vec{\mathbf{f}}_{\texttt{f}}=\vec{0}$, $\vec{\mathbf{f}}_{\texttt{v}}=\vec{0}$, $\mathrm{WP}_{\texttt{bot}}=1$, and $\vec{s}_{\texttt{bot}}=0$ in the main equations [Eqs. (15)–(23)]. A proof for this part will be divided into four cases: (1) $n_{\texttt{f}}=0$ and $n_{\texttt{cap}}=0$, (2) $n_{\texttt{f}}\geq 1$ and $n_{\texttt{cap}}=0$, (3) $n_{\texttt{f}}=0$ and $n_{\texttt{cap}}\geq 1$, and (4) $n_{\texttt{f}}\geq 1$ and $n_{\texttt{cap}}\geq 1$. In each case, the main equations will be simplified by eliminating the terms that are equal to zero. Afterwards, we consider the following pairs of equations: Eqs. (15) and (16), (17) and (18), (22) and (23). For each pair, the types of faults on the 3D structure will be translated to their corresponding types of faults on the 2D plane in order to find matching conditions from Conditions 1–5. Note that the total number of faults of each type will also help in finding the matching conditions, and the total number of faults of all types is at most $d-1$. When the matching conditions are found, we find that some contradictions will happen (assuming that all conditions are satisfied), and this is true for all possible cases. ■

Theorem 1 can make the process of finding CNOT orderings that give a distinguishable fault set less laborious; instead of finding all possible fault combinations arising from the circuits for f, v, and cap generators and checking whether any condition in Definition 3 is satisfied, we just have to check whether all possible fault combinations arising from the circuits for f and cap generators satisfy Conditions 1–5. Note that the number of possible fault combinations of the latter task is much smaller than that of the former task because the total number of generators involved in the latter calculation roughly decreases by half, and the weight of an f generator is half of the weight of its corresponding v generator. After good CNOT orderings for f and cap generators are found, we can find the CNOT orderings of v generators by the constraints imposed by the general configurations for data and flag CNOT gates.

1. *Nonflag circuits for measuring generators of capped color codes in H form of distances 3 and 5*

In the case that all circuits for measuring generators are *nonflag circuits*, we can find good CNOT orderings (which give a distinguishable fault set) for the capped color codes in H form of distances 3 and 5. The circuits and

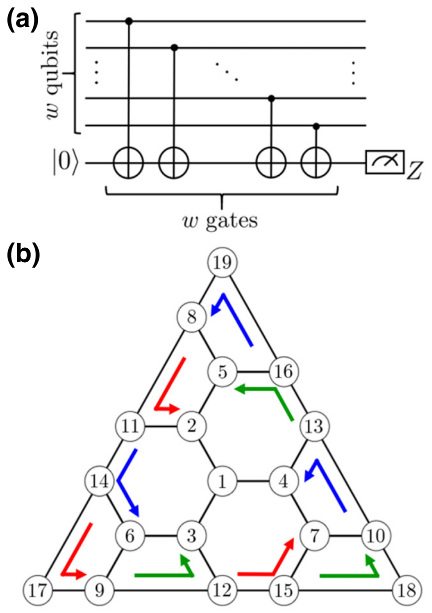


FIG. 12. (a) A nonflag circuit for measuring a generator of the capped color code of distance 5 in H form, where w is the weight of the generator. (b) The orderings of data CNOT gates that give a distinguishable fault set $\mathcal{F}_2$.

CNOT orderings for the code of distance 3 (which is the 3D color code of distance 3) were previously described in Sec. III B. The circuit for measuring a generator of weight w of the code of distance 5 is a nonflag circuit, as shown in Fig. 12(a), and the orderings of data CNOT gates for f and cap generators are given in Fig. 12(b). In Fig. 12(b), for each f generator, the qubits on which data CNOT gates act start from the tail of an arrow and then proceed counterclockwise, and the ordering of data CNOT gates for the cap generator is in numerical order, i.e., (0,1,2,...,19), following the qubit labels in the diagram. Meanwhile, the ordering of data CNOT gates for each v generator can be obtained from its corresponding f generator using the sawtooth configuration (see Sec. III B and Fig. 4 for more details).

The aforementioned results for the codes of distances 3 and 5 are found by manually picking the CNOT ordering for each f or cap generator, and then using a computer simulation to verify that Conditions 1–5 are satisfied. However, searching for good CNOT orderings using this procedure might not be efficient when d is large. We point out that in the case that all circuits for measuring generators are nonflag circuits, it is still not known whether good CNOT orderings exist for $d\geq 7$. Fortunately, we can prove analytically that if all circuits for measuring generators are *flag circuits* of a particular form, it is always possible to obtain a distinguishable fault set for a capped color code in H form of *any distance*.

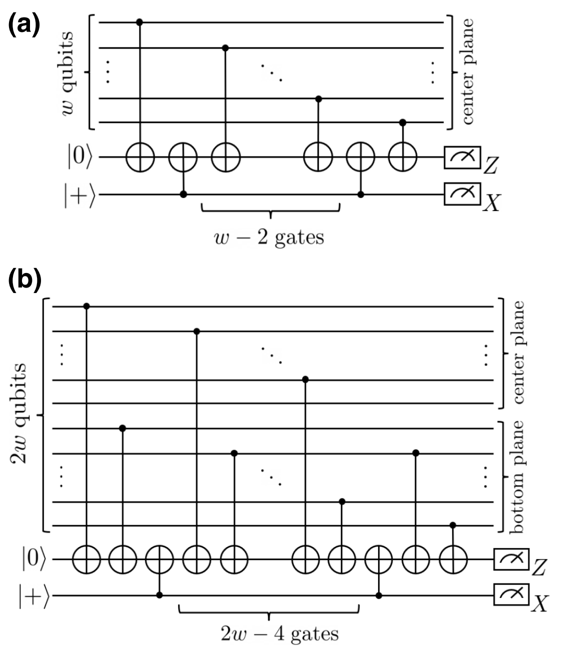


FIG. 13. (a) A flag circuit with one flag ancilla for measuring an f or a cap generator of weight w. (b) A flag circuit with one flag ancilla for measuring a v generator of weight $2w$.

2. Flag circuits for measuring generators of a capped color code in H form of any distance

Here we show that there exist flag circuits for measuring generators of a capped color code in H form of any distance that can give a distinguishable fault set. First, assume that the circuit for measuring an f and a cap generator of weight w is a flag circuit with one flag ancilla similar to the circuit in Fig. 13(a), and that the circuit for measuring a v generator is a flag circuit with one flag ancilla similar to the circuit in Fig. 13(b) (which follows the general configurations of data and flag CNOT gates).

Next, let us consider Eqs. (15)–(23). A nontrivial logical operator of a capped color code in H form with trivial flags happens whenever $s_{\mathtt{cap}} = 0$, $\vec{s}_{\mathtt{f}} = \vec{0}$, $\vec{s}_{\mathtt{v}} = \vec{0}$, $\mathrm{WP}_{\mathtt{tot}} = 1$, $\vec{\mathbf{f}}_{\mathtt{cap}} = \vec{0}$, $\vec{\mathbf{f}}_{\mathtt{f}} = \vec{0}$, $\vec{\mathbf{f}}_{\mathtt{v}} = \vec{0}$, $\mathrm{WP}_{\mathtt{bot}} = 1$, and $\vec{s}_{\mathtt{bot}} = 0$. This means that a nontrivial logical operator of a capped color code in H form [constructed from CCC(d)] occurs if and only if (1) the combined data error on the bottom plane ($\mathbf{E}_{\mathtt{off}}$) is a nontrivial logical operator of the 2D color code of distance d with trivial flags, and either (2.a) $n_0 = 0$ and the combined data error on the center plane ($\mathbf{E}_{\mathtt{on}}$) is a stabilizer of the 2D color code of distance d with trivial flags, or (2.b) $n_0 = 1$ and the combined data error on the center plane ($\mathbf{E}_{\mathtt{on}}$) is a nontrivial logical operator of the 2D color code of distance d with trivial flags. For this reason, if we can show that there is no fault combination from up to $d-1$ faults that can cause a nontrivial logical operator of the 2D color code of distance d with trivial flags on the bottom plane, then a nontrivial logical operator of the capped color code in H form [constructed from CCC(d)] with trivial flags cannot happen, meaning that the fault set $\mathcal{F}_t$ is distinguishable.

Observe that faults that can contribute to $\mathbf{E}_{\mathtt{off}}$ are $\mathtt{q}_{\mathtt{off}}$, v, and v* faults only. Moreover, from the flag circuit for a v generator in Fig. 13(b), a single fault of v or v* type will give a trivial flag only when the part of the corresponding data error on the bottom plane has weight ≤ 1. This fact leads to the following claim.

Claim 1. *Suppose that* v *generators are measured using flag circuits with one flag ancilla similar to the circuit in Fig. 13(b).*

1. *If there is exactly one fault during a measurement of generator v_i^z and the bit of the flag vector corresponding to v_i^z is zero, then the data error on the bottom plane has weight* 0 *or* 1. *In this case, the data error on the bottom plane from one fault of* v *(or* v*) *type is similar to some data error from zero or one fault of* $\mathtt{q}_{\mathtt{off}}$ *type.*
2. *If there are exactly two faults during measurements of the same generator v_i^z (possibly on different rounds) and the bit of the cumulative flag vector corresponding to v_i^z is zero, then the combined data error on the bottom plane has weight* $0, 1, 2,$ *or* 3 *(up to a multiplication of v_i^z). The combined data error of weight* $0, 1,$ *or* 2 *on the bottom plane from two faults of* v *(or* v*) *type on the same generator is similar to some combined data error from zero, one, or two faults of* $\mathtt{q}_{\mathtt{off}}$ *type. The case that the combined data error on the bottom plane of weight* 3 *arising from two faults of* v *(or* v*) *type on the same generator is the only case that the weight of the combined data error on the bottom plane is greater than the number of faults.*
3. *If there are three or more faults during measurements of the same generator v_i^z (possibly from different rounds) and the bit of the cumulative flag vector corresponding to v_i^z is zero, then the combined data error on the bottom plane has weight* $0, 1, 2,$ *or* 3 *(up to a multiplication of v_i^z) and is similar to some combined data error from zero, one, two, or three faults of* $\mathtt{q}_{\mathtt{off}}$ *type.*

Claim 1 will be used later to prove that a nontrivial logical operator of the 2D color code of distance d with trivial flags cannot happen on the bottom plane.

Because the ordering of CNOT gates for each v generator is related to its corresponding f generator, the problem of finding CNOT orderings for a 3D structure that give a distinguishable fault set can be simplified to the problem of finding CNOT orderings on a 2D plane. In particular, since we are now considering the bottom plane only, $\mathtt{f}_{\mathtt{2D}}$ faults on the 2D plane correspond to both v and v* faults on the

3D structure, while $\mathtt{q}_{2\mathrm{D}}$ faults on the 2D plane correspond to $\mathtt{q}_{\mathtt{off}}$ faults on the 3D structure.

A fault set $\mathcal{F}_t$ is distinguishable if the following condition is satisfied.

Condition 6. *For any fault combination on the 2D plane that satisfies $n_{\mathtt{q}_{2\mathrm{D}}} + n_{\mathtt{f}_{2\mathrm{D}}} \le d-1$, $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is not a nontrivial logical operator or the cumulative flag vector is not zero; equivalently, at least one of the following conditions is satisfied:*

1. $\sum \vec{q}_{2\mathrm{D}} + \sum \vec{p}_{\mathtt{f}_{2\mathrm{D}}} \neq 0 \mod 2,$
2. $n_{\mathtt{q}_{2\mathrm{D}}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}_{2\mathrm{D}}}) \neq 1 \mod 2,$
3. $\sum \vec{f}_{\mathtt{f}_{2\mathrm{D}}} \neq 0 \mod 2.$

Surprisingly, using the flag circuits with one flag ancilla as shown in Figs. 13(a) and 13(b) to measure the generators of a capped color code in H form, Condition 6 is satisfied regardless of the orderings of data CNOT gates of $\mathtt{f}$ generators (as long as the CNOT orderings of $\mathtt{v}$ generators follow the general configurations of data and flag CNOT gates). And because we are considering faults on the (simplified) 2D plane, the fact that Condition 6 is satisfied regardless of the orderings of data CNOT gates in the flag circuits is also applicable to a 2D color code of any distance as well. This can be restated in the following theorem.

Theorem 2. *Suppose that the generators of a 2D color code of distance d are measured using the flag circuits with one flag ancilla as displayed in Fig. 13(a). Then, there is no fault combination arising from $d-1$ faults whose combined data error is a nontrivial logical operator and the cumulative flag vector is zero (i.e., Condition 6 is satisfied), regardless of the orderings of data CNOT gates in the flag circuits.*

Theorem 2 has been proved in Ref. [24], where the circuit in Fig. 13(a) is a 1-flag circuit according to the definition in Ref. [22]. Here we also provide an alternative proof of Theorem 2 that is tailored to the notation being used throughout this work, so that the paper becomes self-contained. We also believe that our proof technique using the relationship between faults and error weights would be useful for finding proper CNOT orderings for other families of codes.

Proof of Theorem 2. Assume by contradiction that Condition 6 is not satisfied; i.e., there exists a fault combination from $d-1$ faults that gives a nontrivial logical operator with trivial flags. For such a fault combination, the syndrome of $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is zero, the total weight of $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is odd, and the cumulative flag vector $\sum \vec{f}_{\mathtt{f}_{2\mathrm{D}}}$ is zero. From the structure of the flag circuit in Fig. 13(a), a single fault of $\mathtt{f}_{2\mathrm{D}}$ type will give a trivial flag only when the corresponding data error has weight ≤ 1. Similar to Claim 1 for faults of $\mathtt{v}$ and $\mathtt{v}^*$ type discussed previously, the only case that faults of $\mathtt{f}_{2\mathrm{D}}$ type cannot be considered as faults of $\mathtt{q}_{2\mathrm{D}}$ type of the same or smaller number is the case that, for each generator f_i^z of the 2D color code, there are exactly two faults during the generator measurements (on the same or different rounds) that lead to the combined data error of weight 3 (up to a multiplication of f_i^z). For this reason, we assume that, for each generator f_i^z, there are either no faults or exactly two faults during the measurements.

Let (n_f, n_q) denote the case that a fault combination arises from *exactly* n_f faults of $\mathtt{f}_{2\mathrm{D}}$ type and *no more than* n_q faults of $\mathtt{q}_{2\mathrm{D}}$ type (where $n_f + n_q = d-1$). We show that in any case with even n_f [i.e., $(0, d-1), (2, d-3), \ldots, (d-1, 0)$], $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ cannot be a nontrivial logical operator.

Case $(0, d-1)$. Because the 2D color code has distance d and the total weight of $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}}$ is at most $d-1$, $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}}$ cannot be a nontrivial logical operator.

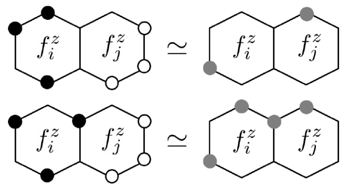
Case $(2, d-3)$. Suppose that a pair of $\mathtt{f}_{2\mathrm{D}}$ faults causes a weight-3 error on the supporting qubits of generator f_i^z. Consider the following cases.

1. If there are an even number of $\mathtt{q}_{2\mathrm{D}}$ faults on the supporting qubits of f_i^z, then the syndrome bit s_i^x corresponding to generator f_i^x is not zero. That is, $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is not a nontrivial logical operator.
2. If there are an odd number of $\mathtt{q}_{2\mathrm{D}}$ faults on the supporting qubits of f_i^z, then the total weight of the error on supporting qubits of f_i^z is 0 or 2 (the total weight is even and no more than 3 up to a multiplication of f_i^z). Since two $\mathtt{f}_{2\mathrm{D}}$ faults and one or more $\mathtt{q}_{2\mathrm{D}}$ faults give an error of weight no more than 2, this case is covered by the $(0, d-1)$ case, in which a nontrivial logical operator cannot occur.

Thus, $\mathbf{E}_{\mathtt{q}_{2\mathrm{D}}} \cdot \mathbf{E}_{\mathtt{f}_{2\mathrm{D}}}$ is not a nontrivial logical operator in the $(2, d-3)$ case.

Case (n_f, n_q) *with* $n_f \ge 4$ *and* $n_f + n_q = d-1$. Consider the following cases.

1. The case in which there are two pairs of $\mathtt{f}_{2\mathrm{D}}$ faults that occur on adjacent generators f_i^z and f_j^z, and each pair leads to an error of weight 3 on the supporting qubits of each generator. We can always make these two errors of weight 3 overlap by multiplying each error with f_i^z (or f_j^z); see the examples below.



As a result, the total weight of these two errors becomes 2 or 4. Since four $\mathtt{f}_{\mathtt{2D}}$ faults give an error of weight no more than 4, this case is covered by the (n_f-4, n_q+4) case. We can repeat this reduction process until there are no pairs of faults that occur on adjacent generators.

2. The case in which there are no pairs of $\mathtt{f}_{\mathtt{2D}}$ faults that occur on adjacent generators. Suppose that a single pair of $\mathtt{f}_{\mathtt{2D}}$ faults causes a weight-3 error on the supporting qubits of generator f_i^z.
   (a) If there are an even number of $\mathtt{q}_{\mathtt{2D}}$ faults on the supporting qubits of f_i^z, then the syndrome bit s_i^x corresponding to generator f_i^x is not zero. That is, $\mathbf{E}_{\mathtt{q}_{\mathtt{2D}}}\cdot\mathbf{E}_{\mathtt{f}_{\mathtt{2D}}}$ is not a nontrivial logical operator.
   (b) If there are an odd number of $\mathtt{q}_{\mathtt{2D}}$ faults on the supporting qubits of f_i^z, then the total weight of the error on supporting qubits of f_i^z is 0 or 2 (up to a multiplication of f_i^x). Since two $\mathtt{f}_{\mathtt{2D}}$ faults and one or more $\mathtt{q}_{\mathtt{2D}}$ faults give an error of weight no more than 2, this case is covered by the (n_f-2, n_q+2) case.

By induction, a nontrivial logical operator cannot occur in any case with $n_f\geq 4$ and $n_f+n_q=d-1$.

Therefore, there is no fault combination from $d-1$ faults that gives the zero cumulative flag vector and a nontrivial logical operator on the 2D color code. ■

From Theorem 2, it is always possible to obtain a distinguishable fault set $\mathcal{F}_t$ for a 2D color code of any distance (thus, fault-tolerant protocols for error correction, measurement, and state preparation described in Sec. V are applicable).

Now let us consider the capped color code in H form. Because there is no fault combination from $d-1$ faults that can cause a nontrivial logical operator of the 2D color code with trivial flags on the bottom plane, a nontrivial logical operator of the capped color code in H form with trivial flags cannot occur from $d-1$ faults. By Proposition 1, this implies that the fault set $\mathcal{F}_t$ is distinguishable. The result can be summarized in the following theorem.

Theorem 3. *Let $\mathcal{F}_t$ be the fault set corresponding to circuits for measuring $\mathtt{f}, \mathtt{v}$, and $\mathtt{cap}$ generators of the capped color code in H form constructed from* CCC(d) *[where $t=(d-1)/2$, $d=3,5,7,\ldots$], and suppose that the general configurations of* CNOT *gates for $\mathtt{f}$, $\mathtt{v}$, and $\mathtt{cap}$ generators are imposed, and that the circuits for each pair of X-type and Z-type generators use the same* CNOT *ordering. Also, let circuits for measuring $\mathtt{f}$ and $\mathtt{cap}$ generators be flag circuits with one flag ancilla similar to the circuit in Fig. 13(a), and let circuits for measuring $\mathtt{v}$ generators be flag circuits with one flag ancilla similar to the circuit in Fig. 13(b). Then, $\mathcal{F}_t$ is distinguishable.*

(We can also see that whenever Condition 6 is satisfied, Conditions 1–5 are also satisfied. This leads to a distinguishable fault set by Theorem 1.)

The fault-tolerant protocols for error correction, measurement, and state preparation in Sec. V are applicable to a capped color code in H form of any distance whenever the fault set is distinguishable. Note that the protocols for capped color codes in H form of distances 3 and 5 need only one ancilla in total, while the protocols for code of distance 7 or higher need only two ancillas in total (assuming that the ancillas can be reused).

In addition, the CNOT orderings that work for capped color codes in H form will work for recursive capped color codes in H form. That is, for a recursive capped color code in H form of distance $d=2t+1$, the fault set $\mathcal{F}_t$ is distinguishable if the following statements hold:

1. the $\mathtt{f}$ and $\mathtt{v}$ operators on the $(j-1)$th and the jth layers of the recursive capped color code are measured using the CNOT orderings for the $\mathtt{f}$ and $\mathtt{v}$ operators of a capped color code in H form of distance j $(j=3,5,\ldots,d)$ that give a distinguishable fault set, and
2. the $\mathtt{cap}$ operator on the $(j-2)$th and the $(j-1)$th layers of the recursive capped color code is measured using the CNOT ordering for the $\mathtt{cap}$ operator of a capped color code in H form of distance j $(j=3,5,\ldots,d)$ that give a distinguishable fault set [where an operator on $\mathtt{q}_0$ of the capped color code is replaced by operators on all qubits on the $(j-2)$th layer of the recursive capped color code].

The orderings above work because the recursive capped color code in H form of distance d is obtained by encoding the top qubit $(\mathtt{q}_0)$ of the capped color code in H form of distance d by the recursive capped color code in H form of distance $d-2$. FTEC protocols for a recursive capped color code in H form are similar to conventional FTEC protocols for a concatenated code; we start from correcting errors on the innermost code and then proceed outwards. Other fault-tolerant protocols for a recursive capped color code will also use similar ideas.

V. FAULT-TOLERANT PROTOCOLS

So far, we have considered capped and recursive capped color codes in H form, and derived Theorems 1–3 that help us find CNOT orderings for the circuits for measuring the code generators such that the corresponding fault set is distinguishable. In this section, we show that whenever the fault set is distinguishable, a fault-tolerant protocol can be constructed. We first state the definitions of fault-tolerant gadgets in Sec. V A, which are a bit different from conventional definitions originally proposed by Aliferis, Gottesman, and Preskill (AGP) [10]. Afterwards, we develop several fault-tolerant protocols for a capped or a recursive capped color code whose circuits for measuring generators give a distinguishable fault set, including a FTEC protocol (Sec. V B), fault-tolerant measurement (FTM) and fault-tolerant state preparation (FTP) protocols (Sec. V C), transversal Clifford gates (Sec. V D), and a fault-tolerant protocol for logical T-gate implementation (Sec. V E).

A. Redefining fault tolerance

When a fault set $\mathcal{F}_t$ is distinguishable, all possible errors of any weight arising from up to t faults can be accurately identified (up to a multiplication of some stabilizer) using their syndromes and cumulative flag vectors obtained from perfect subsequent syndrome measurements. Therefore, all possible errors arising from up to t faults are correctable. However, one should be aware that faults can happen anywhere in an EC protocol, including the locations in the subsequent syndrome measurements. Our goal is to construct a protocol that is *fault tolerant*; vaguely speaking, if an input state to an EC protocol has some error, we want to make sure that the output state is the same logical state as the input, and if output state has any error, the error must not be "too large".

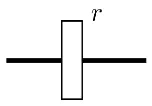
What does it mean for the output error to be not too large? The general idea is that if an output error of a single round of the protocol becomes an input error of the next round of the protocol, the error should still be correctable by the latter round. Aliferis *et al.* [10] proposed that the weight of the output error from a fault-tolerant protocol should be no more than the number of total faults that occurred during the protocol. However, it should be noted that, for an $[[n,k,d]]$ code that can correct errors up to weight $\tau = \lfloor (d-1)/2 \rfloor$ and is not a perfect code (or not a perfect CSS code) [83], the idea of correctable errors can be extended to some errors of weight more than τ. For example, if the code being used is a nonperfect code of distance 3, there will be some error E of weight more than 1 whose syndrome $\vec{s}(E)$ is different from those of errors of weight 1. If no other error E' has the same syndrome as E in the set of correctable errors, then in this case E is also correctable in the sense that we can perform an error correction by applying $E^\dagger$ every time we obtained the syndrome $\vec{s}(E)$. In this section, we "refine" the idea of high-weight error correction and "redefine" fault tolerance using the notion of distinguishable fault set.

We start by stating conventional definitions of fault-tolerant gadgets proposed by Aliferis *et al.* [10]; then we give the revised version of the same definitions. Recall that τ denotes the weight of errors that a stabilizer code can correct and t denotes the number of faults. The first two definitions are the definitions of an r filter and an ideal decoder, which are the main tools for describing the properties of fault-tolerant gadgets.

Definition 5 (r filter—AGP version): Let $T(S)$ be the coding subspace defined by the stabilizer group S. An r filter is the projector onto the subspace spanned by

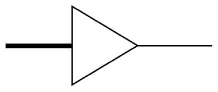
$$\{E|\bar{\psi}\rangle; |\bar{\psi}\rangle \in T(S), \text{ the weight of } E \text{ is at most } r\}. \quad (24)$$

An r filter in circuit form is



where a thick line represents a block of code.

Definition 6 (Ideal decoder—AGP version): Let $\tau = \lfloor (d-1)/2 \rfloor$, where d is the code distance. An ideal decoder is a gadget that can correct any error of weight up to τ and map an encoded state $|\bar{\psi}\rangle$ on a code block to the corresponding (unencoded) state $|\psi\rangle$ on a single qubit without any fault. An ideal decoder in circuit form is



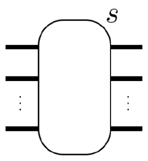
where a thick line represents a block of code and a thin line represents a single qubit.

The intuition behind the definitions of these two gadgets is as follows. If an input state of an r filter differs from a codeword by an error of weight $\leq r$, then the output state will also differ from the same codeword by an error of weight $\leq r$. However, if the input state has an error of weight $> r$, then the input and output states may correspond to different ideal codewords (i.e., they may be ideally decoded to different unencoded states). An ideal decoder is a gadget that guarantees that the output (unencoded) state and the input (encoded) state will be logically the same whenever the input state has an error of weight no more than τ.

(Note that an r filter is a linear, completely positive map, but it is not trace preserving; an r filter cannot be physically implemented. In the definitions of fault-tolerant gadgets to be described, r filters will be used as mathematical objects to express circuit identities that must hold when the weight of input or output errors and the number of faults are restricted. When each identity holds, both sides of the equation give the same output, including normalization, for the same input state, but the trace of the output might not be one.)

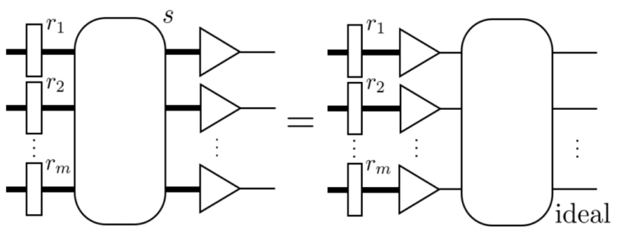
Using the definitions of the r filter and ideal decoder, a fault-tolerant gate (FTG) gadget and FTEC gadget can be defined as follows.

Definition 7 (Fault-tolerant gate gadget—AGP version): A *gate gadget* with s faults simulating an ideal m-qubit gate is represented by the following picture:

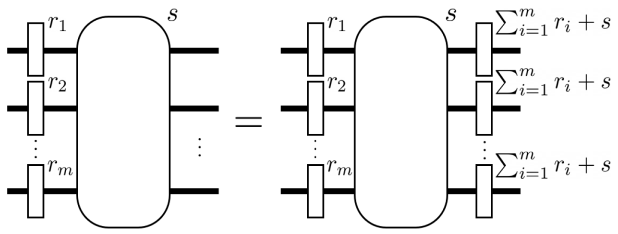


where each thick line represents a block of code. Let $t \leq \lfloor (d-1)/2 \rfloor$. A gate gadget is *t-fault tolerant* if it satisfies both of the following properties.

1. Gate correctness property (GCP): whenever $\sum_{i=1}^{m} r_i + s \leq t$,

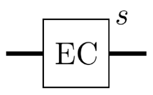


2. Gate error propagation property (GPP): whenever $\sum_{i=1}^{m} r_i + s \leq t$,



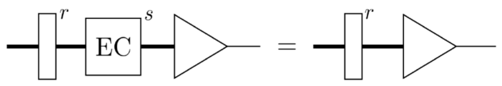
Here the r-filter and the ideal decoder are as defined in Definitions 5 and 6.

Definition 8 (Fault-tolerant error-correction gadget—AGP version): An *error-correction gadget* with s faults is represented by the following picture:

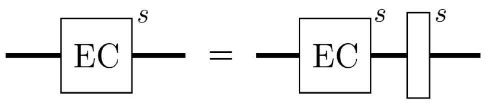


where a thick line represents a block of code. Let $t \leq \lfloor (d-1)/2 \rfloor$. An error-correction gadget is *t-fault tolerant* if it satisfies both of the following properties.

1. Error-correction correctness property (ECCP): whenever $r + s \leq t$,



2. Error-correction recovery property (ECRP): whenever $s \leq t$,

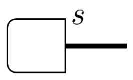


Here the r filter and the ideal decoder are as defined in Definitions 5 and 6.

When a FTG gadget satisfies both properties in Definition 7, it is guaranteed that whenever the weight of the input error plus the number of faults is no more than t, (1) the operation of a FTG gadget on an encoded state will be similar to the operation of its corresponding quantum gate on an unencoded state, and (2) an output state of a FTG gadget will have an error of weight no more than t (which is also $\leq \tau$). Meanwhile, the two properties of a FTEC gadget in Definition 8 guarantee that (1) the output and the input states of a FTEC gadget are logically the same whenever the weight of the input error plus the number of faults is no more than t, and (2) the weight of the output error of a FTEC gadget is no more than the number of faults whenever the number of faults is at most t, regardless of the weight of the input error.

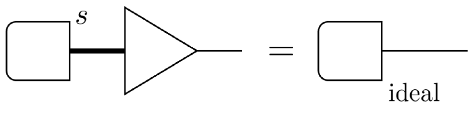
FTP and FTM gadgets, which are special cases of FTG gadget, can be defined as follows.

Definition 9 (Fault-tolerant state preparation gadget—AGP version): A *state preparation gadget* with s faults is represented by the following picture:

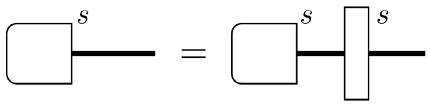


where a thick line represents a block of code. Let $t \leq \lfloor (d-1)/2 \rfloor$. A state preparation gadget is *t-fault tolerant* if it satisfies both of the following properties.

1. Preparation correctness property (PCP): whenever $s \leq t$,

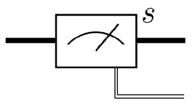


2. Preparation error propagation property (PPP): whenever $s \leq t$,



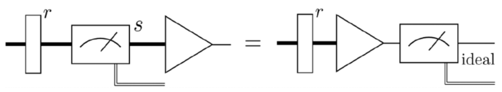
Here the r filter and the ideal decoder are defined as in Definitions 5 and 6.

Definition 10 (Fault-tolerant (nondestructive) measurement gadget—AGP version): A *(nondestructive) measurement gadget* with s faults is represented by the following picture:

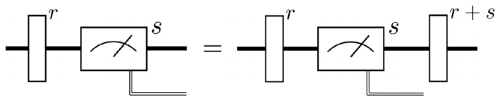


where a thick line represents a block of code. Let $t \leq \lfloor (d-1)/2 \rfloor$. A (nondestructive) measurement gadget is *t-fault tolerant* if it satisfies both of the following properties.

1. Measurement correctness property (MCP): whenever $r + s \leq t$,



2. Measurement error propagation property (MPP): whenever $r + s \leq t$,



Here the r filter and the ideal decoder are defined as in Definitions 5 and 6.

The meanings of the properties of FTP and FTM gadgets are similar to the meanings of the properties of a FTG gadget as previously explained.

From Definitions 7–10, we can see that an action of a fault-tolerant gadget is guaranteed in the circumstance that the weight of the input error r and the number of faults that occurred in the gadget s satisfy some condition. Now, a question that arises is: what will happen if the input error has weight greater than $\tau = \lfloor (d-1)/2 \rfloor$, which is the weight of errors that a code can correct? By Definition 3, we know that if a fault set $\mathcal{F}_t$ is distinguishable, possible errors arising from up to t faults in an EC protocol [where $t \leq \lfloor (d-1)/2 \rfloor$] can be distinguished using their corresponding syndromes or cumulative flag vectors, regardless of the error weights. Would it be more natural if the definitions of fault-tolerant gadgets depend on the *number of faults* related to an input error, instead of the *weight* of an input error? In this work, we try to modify the definitions of fault-tolerant gadgets and rewrite them using the notion of distinguishable fault set.

To modify the definitions of fault-tolerant gadgets proposed in Ref. [10], first let us define a distinguishable error set as follows.

Definition 11 (Distinguishable error set): Let $\mathcal{F}_r$ be a distinguishable fault set, and let $\mathcal{F}_r|_{\vec{\mathbf{f}}=0}$ be a subset of $\mathcal{F}_r$ defined as

$$\mathcal{F}_r|_{\vec{\mathbf{f}}=0} = \{\Lambda \in \mathcal{F}_r;\ \vec{\mathbf{f}} \text{ of } \Lambda \text{ is zero}\}. \tag{25}$$

A *distinguishable error set* $\mathcal{E}_r$ corresponding to $\mathcal{F}_r$ is

$$\mathcal{E}_r = \{\mathbf{E} \text{ of } \Lambda \in \mathcal{F}_r|_{\vec{\mathbf{f}}=0}\}. \tag{26}$$

If $\mathcal{F}_r$ is distinguishable, $\mathcal{F}_r|_{\vec{\mathbf{f}}=0}$ is also distinguishable since all pairs of fault combinations in $\mathcal{F}_r|_{\vec{\mathbf{f}}=0}$ also satisfy the conditions in Definition 3. Moreover, because all fault combinations in $\mathcal{F}_r|_{\vec{\mathbf{f}}=0}$ correspond to the zero cumulative flag vector, we find that, for any pair of errors in $\mathcal{E}_r$, the errors either have different syndromes or are logically equivalent (up to a multiplication of a stabilizer). For this reason, we can safely say that $\mathcal{E}_r$ is a set of correctable errors.

Because the set of correctable errors is now expanded, the definitions of the r filter and ideal decoder can be revised as follows.

Definition 12 (r filter—revised version): Let $T(S)$ be the coding subspace defined by the stabilizer group S, and let $\mathcal{E}_r$ be the distinguishable error set corresponding to a distinguishable fault set $\mathcal{F}_r$. An r filter is the projector onto subspace spanned by

$$\{E|\bar{\psi}\rangle; |\bar{\psi}\rangle \in T(S), E \in \mathcal{E}_r\}. \tag{27}$$

A (revised) r filter in circuit form is similar to that illustrated in Definition 5.

Definition 13 (Ideal decoder—revised version): Let $\mathcal{E}_t$ be the distinguishable error set corresponding to a distinguishable fault set $\mathcal{F}_t$, where $t \leq \lfloor (d-1)/2 \rfloor$ and d is the code distance. An ideal decoder is a gadget that can correct any error in $\mathcal{E}_t$ and map an encoded state $|\bar{\psi}\rangle$ on a code block to the corresponding (unencoded) state $|\psi\rangle$ on a single qubit without any faults. A (revised) ideal decoder in circuit form is similar to that illustrated in Definition 6.

Using the revised definitions of the r filter and ideal decoder, fault-tolerant gadgets can be defined as follows.

Definition 14 (Fault-tolerant gadgets—revised version): Let $t \leq \lfloor (d-1)/2 \rfloor$. Fault-tolerant gadgets are defined as follows.

1. A *gate gadget* is *t-fault tolerant* if it satisfies both of the properties in Definition 7, except that the r filter

and ideal decoder are defined as in Definitions 12 and 13.
2. An *error-correction gadget* is *t-fault tolerant* if it satisfies both of the properties in Definition 8, except that the r filter and ideal decoder are defined as in Definitions 12 and 13.
3. A *state preparation gadget* is *t-fault tolerant* if it satisfies both of the properties in Definition 9, except that the r filter and ideal decoder are defined as in Definitions 12 and 13.
4. A *(nondestructive) measurement gadget* is *t-fault tolerant* if it satisfies both of the properties in Definition 10, except that the r filter and ideal decoder are defined as in Definitions 12 and 13.

The revised definitions of fault-tolerant gadgets in circuit form may look very similar to the old definitions proposed in Ref. [10], but the meanings are different: the conditions in the revised definitions depend on the number of faults that can cause an input or an output error, instead of the weight of an input or an output error. Roughly speaking, this means that (1) a fault-tolerant gadget is allowed to produce an output error of weight greater than τ [where $\tau = \lfloor (d-1)/2 \rfloor$], and (2) a fault-tolerant gadget can work perfectly even though the input error has weight greater than τ, as long as the input or the output error is similar to an error caused by no more than $t \leq \tau$ faults. Because the revised definitions of the r filter and ideal decoder are more general than the old definitions, we expect that a gadget that satisfies one of the old definitions of fault-tolerant gadgets (Definitions 7–10) will also satisfy the new definitions in Definition 14. Note that the revised definitions are based on the fact that a fault set relevant to a gadget is distinguishable, that is, whether the gadgets are fault tolerant depends on the way they are designed.

In a special case where the code being used is a CSS code and possible X-type and Z-type errors have the same form, the definition of a distinguishable error set can be further extended as follows.

Definition 15 (Distinguishable error set (for a special family of CSS codes)): Let $\mathcal{F}_r$ be a distinguishable fault set, and let $\mathcal{F}_r|_{\vec{\mathbf{f}}=0}$ be a subset of $\mathcal{F}_r$ defined as

$$\mathcal{F}_r|_{\vec{\mathbf{f}}=0} = \{\Lambda \in \mathcal{F}_r;\ \vec{\mathbf{f}} \text{ of } \Lambda \text{ is zero}\}. \tag{28}$$

A *distinguishable-X error set* $\mathcal{E}_r^x$ and a *distinguishable-Z error set* $\mathcal{E}_r^z$ corresponding to $\mathcal{F}_r$ are

$$\mathcal{E}_r^x = \{\mathbf{E} \text{ of } \Lambda \in \mathcal{F}_r|_{\vec{\mathbf{f}}=0}; \mathbf{E} \text{ is an } X\text{-type error}\}, \tag{29}$$

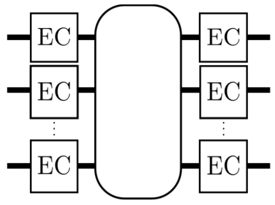
$$\mathcal{E}_r^z = \{\mathbf{E} \text{ of } \Lambda \in \mathcal{F}_r|_{\vec{\mathbf{f}}=0}; \mathbf{E} \text{ is a } Z\text{-type error}\}. \tag{30}$$

For a CSS code in which the elements of $\mathcal{E}_r^x$ and $\mathcal{E}_r^z$ have a similar form, a *distinguishable error set* $\mathcal{E}_r$ corresponding to $\mathcal{F}_r$ is defined as

$$\mathcal{E}_r = \{E_x \cdot E_z; E_x \in \mathcal{E}_r^x, E_z \in \mathcal{E}_r^z\}. \tag{31}$$

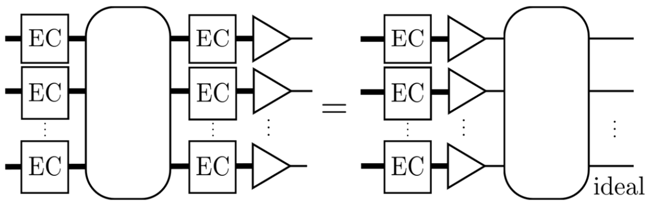
Since a CSS code can detect and correct X-type and Z-type errors separately, here we modify the definition of a distinguishable error set for a CSS code in which $\mathcal{E}_r^x$ and $\mathcal{E}_r^z$ are in the same form so that more Y-type errors are included in $\mathcal{E}_r$. For example, suppose that $t=2$, each of $XXXX$ and $ZZZZ$ can be caused by two faults, and that $YYYY$ can be caused by four faults. By the old definition (Definition 11), we say that $XXXX$ and $ZZZZ$ are in $\mathcal{E}_2$, and $YYYY$ is in $\mathcal{E}_4$ but not in $\mathcal{E}_2$. In contrast, by Definition 15, we say that $XXXX$, $YYYY$, and $ZZZZ$ are all in $\mathcal{E}_2$. This modification will give more flexibility when developing a fault-tolerant gadget for this special kind of CSS code, e.g., a transversal S gate that produces an output error $YYYY$ from an input error $XXXX$ still satisfies the properties in Definition 14 when a distinguishable fault set is defined as in Definition 15.

When performing a fault-tolerant quantum computation, FTEC gadgets will be used repeatedly in order to reduce the error accumulation during the computation. Normally, FTEC gadgets will be placed before and after other gadgets (FTG, FTP, or FTM gadgets). A group of gadgets including a FTG gadget, leading EC gadgets (the FTEC gadgets before the FTG gadget), and trailing EC gadgets (FTEC gadgets after the FTG gadget) as shown below is called an *extended rectangle at level* 1 or 1-*exRec*:



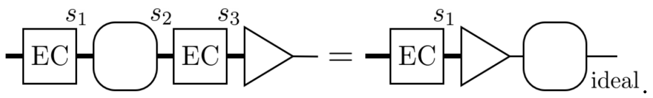
(A 1-exRec of a FTP or FTM gadget is defined similarly to a 1-exRec of a FTG gadget, except that there is no leading gadget in a FTP gadget.) We say that a 1-exRec is *good* if the total number of faults in a 1-exRec is no more than t. Using the revised definitions of fault-tolerant gadgets in Definition 14, a revised version of the exRec-Cor lemma at level 1, originally proposed in Ref. [10], can be obtained.

Lemma 3 (ExRec-Cor lemma at level 1—revised version). *Suppose that all gadgets are t-fault tolerant according to Definition 14. If a 1-exRec is good (i.e., a 1-exRec has no more than t faults), then the 1-exRec is correct; that* *is, the following condition is satisfied:*



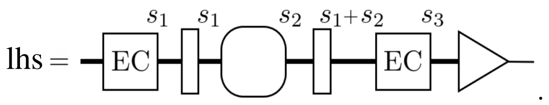
with the r filter and ideal decoder defined as in Definitions 12 and 13.

Proof. Here we focus only on the case that a gate gadget simulates a single-qubit gate. The proofs for the case of a multiple-qubit gate and other gadgets are similar. Suppose that the leading EC gadget, the gate gadget, and the trailing EC gadget in an exRec have s_1, s_2, and s_3 faults, where $s_1 + s_2 + s_3 \leq t$. We show that the following equation holds:

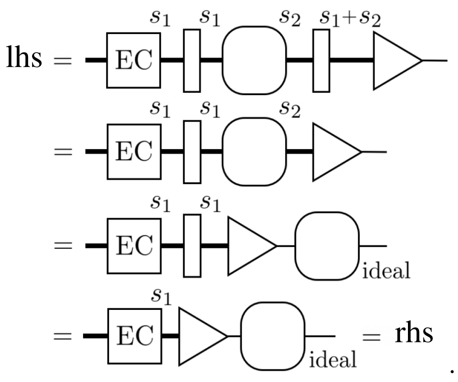


(32)

Because the gate gadget satisfies GPP and the EC gadgets satisfy ECRP, the left-hand side of Eq. (32) is



Using GCP, ECCP, and the fact that an ideal decoder can correct any error in $\mathcal{E}_t$, we obtain



This completes the proof. ■

(Note that both sides of the equation in Lemma 3 are trace-preserving, completely positive maps, even though the r filters introduced during the proof are not trace preserving. This is possible since the total number of faults in a 1-exRec is restricted and all gadgets satisfy Definition 14.)

The revised version of the exRec-cor lemma developed in this work is very similar to the original version in Ref. [10], even though the r filter, the ideal decoder, and the fault-tolerant gadgets are redefined. The exRec-Cor lemma is one of the main ingredients for the proofs of other lemmas and theorems in Ref. [10]. As a result, other lemmas and theorems developed in Ref. [10] are also applicable to our case, including their version of the *threshold theorem* (the proofs of the revised versions of the lemmas and theorems are similar to the proofs presented in Ref. [10], except that Lemma 3 is used instead of the original exRec-Cor lemma). This means that fault-tolerant gadgets satisfying Definition 14 can be used to simulate any quantum circuit, and the logical error rate can be made arbitrarily small if the physical error rate is below some constant threshold value. The main advantage of the revised definitions of fault-tolerant gadgets over the conventional definitions is that high-weight errors are allowed as long as they arise from a small number of faults. These revised definitions can give us more flexibility when developing fault-tolerant protocols.

B. Fault-tolerant error-correction protocol

So far, we have shown that it is possible to redefine the r filter and ideal decoder as in Definitions 12 and 13 using the notions of distinguishable fault set (Definition 3) and distinguishable error set (Definition 11 or Definition 15), and redefine fault-tolerant gadgets as in Definition 14. These revised definitions give us more flexibility when designing fault-tolerant protocols, while ensuring that the simulated circuit constructed from these protocols still work fault-tolerantly. In this section, we construct a FTEC protocol for a capped color code in H form of any distance in which its fault set is distinguishable. Note that having only the FTEC protocol is not enough for general fault-tolerant quantum computation, so we also construct other fault-tolerant protocols that share the same distinguishable fault set with the FTEC protocol for a particular code in Secs. V C–V E.

To construct a FTEC protocol for a capped color code in H form obtained from CCC(d), we first assume that the fault set $\mathcal{F}_t$ [where $t = (d-1)/2$] corresponding to the circuits for measuring the generators of the code is distinguishable, and that the orderings of gates in the circuits for each pair of X-type and Z-type generators are the same. From the fact that $\mathcal{F}_t$ is distinguishable, we can build a list of all possible fault combinations and their corresponding combined error, syndrome of the combined error, and cumulative flag vector. Note that if several fault combinations have the same syndrome and cumulative flag vector, their combined errors are all logically equivalent (from Definition 3).

Let $\vec{\mathbf{s}} = (\vec{\mathbf{s}}_x|\vec{\mathbf{s}}_z)$ be the syndrome obtained from the measurements of X-type and Z-type generators, and let $\vec{\mathbf{f}} =$

$(\vec{\mathbf{f}}_x|\vec{\mathbf{f}}_z)$ be the cumulative flag vector corresponding to the flag outcomes from the circuits for measuring X-type and Z-type generators, where $\vec{\mathbf{f}}$ is accumulated from the first round until the current round. We define the *outcome bundle* $(\vec{\mathbf{s}},\vec{\mathbf{f}})$ to be the collection of $\vec{\mathbf{s}}$ and $\vec{\mathbf{f}}$ obtained during a single round of full syndrome measurement. A FTEC protocol for the capped color code in H form is as follows.

FTEC protocol for a capped color code in H form. During a single round of full syndrome measurement, measure the generators in the following order: measure the v_i^x, then the $f_i{}^x$, then the v_i^z, then the $f_i{}^z$. Perform full syndrome measurements until the outcome bundles $(\vec{\mathbf{s}},\vec{\mathbf{f}})$ are repeated $t+1$ times in a row. Afterwards, do the following.

1. Determine an EC operator F_x using the list of possible fault combinations as follows.
   (a) If there is a fault combination on the list whose syndrome and cumulative flag vector are $(\vec{0}|\vec{\mathbf{s}}_z)$ and $(\vec{\mathbf{f}}_x|\vec{0})$, then F_x is the combined error of such a fault combination. (If there is more than one fault combination corresponding to $(\vec{0}|\vec{\mathbf{s}}_z)$ and $(\vec{\mathbf{f}}_x|\vec{0})$, a combined error of any such fault combinations will work.)
   (b) If none of the fault combinations on the list corresponds to $(\vec{0}|\vec{\mathbf{s}}_z)$ and $(\vec{\mathbf{f}}_x|\vec{0})$, then F_x can be any Pauli-X operator whose syndrome is $(\vec{0}|\vec{\mathbf{s}}_z)$.
2. Determine an EC operator F_z using the list of possible fault combinations as follows.
   (a) If there is a fault combination on the list whose syndrome and cumulative flag vector are $(\vec{\mathbf{s}}_x|\vec{0})$ and $(\vec{0}|\vec{\mathbf{f}}_z)$, then F_z is the combined error of such a fault combination. (If there is more than one fault combination corresponding to $(\vec{\mathbf{s}}_x|\vec{0})$ and $(\vec{0}|\vec{\mathbf{f}}_z)$, a combined error of any such fault combinations will work.)
   (b) If none of the fault combinations on the list corresponds to $(\vec{\mathbf{s}}_x|\vec{0})$ and $(\vec{0}|\vec{\mathbf{f}}_z)$, then F_z can be any Pauli-Z operator whose syndrome is $(\vec{\mathbf{s}}_x|\vec{0})$.
3. Apply $F_x \cdot F_z$ to the data qubits to perform error correction.

To verify that the above EC protocol is fault tolerant according to the revised definition (Definition 14), we have to show that the two properties in Definition 8 are satisfied when the r filter and the ideal decoder are defined as in Definitions 12 and 13 (instead of Definitions 5 and 6) and the distinguishable error set is defined as in Definition 15 (the circuits for X-type and Z-type generators of the capped color code in H form use similar gate orderings). Here we assume that there are no more than t faults during the whole protocol. Therefore, the condition that the outcome bundles are repeated $t+1$ times in a row will be satisfied within $(t+1)^2$ rounds. We divide the analysis into two cases: (1) the case that the last round of the full syndrome measurement has no faults and (2) the case that the last round has some faults.

(1) Because the outcome bundles are repeated $t+1$ times and the last round of the full syndrome measurement has no faults, we know that the outcome bundle of the last round is correct and corresponds to the data error before the error correction in step 3. Let E_{in} be the input error and E_a be the combined error of a fault combination arising from the s_a faults, where $s_a \leq t$. The error on the data qubits before step 3 is $E_a \cdot E_{\text{in}}$. First, consider the case that E_{in} is in $\mathcal{E}_r$ (defined in Definition 15), where $r + s_a \leq t$. Both E_{in} and E_a can be separated into X and Z parts. We find that the X part of E_{in} is in $\mathcal{E}_r^x$ (which is derived from $\mathcal{F}_r|_{\vec{\mathbf{f}}=0}$). Thus, the X part of $E_a \cdot E_{\text{in}}$ is the combined error of X type of some fault combination in $\mathcal{F}_{r+s_a}$. Similarly, the Z part of E_{in} is in $\mathcal{E}_r^z$, and the Z part of $E_a \cdot E_{\text{in}}$ is the combined error of Z type of some fault combination in $\mathcal{F}_{r+s_a}$. By picking EC operators F_x and F_z as in steps 1(a) and 2(a), step 3 can completely remove the data error. Thus, both ECCP and ECRP in Definition 8 are satisfied. On the other hand, if E_{in} is not in $\mathcal{E}_r$, where $r + s_a \leq t$, the X part or the Z part of $E_a \cdot E_{\text{in}}$ might not correspond to any fault combination in $\mathcal{F}_t$. In this case, F_x or F_z will be picked as in step 1(b) or 2(b). Because the X part (or the Z part) of $E_a \cdot E_{\text{in}}$ and F_x (or F_z) have the same syndrome no matter how we pick F_x (or F_z), the output state after step 3 is a valid codeword, but it may or may not be logically the same as the input state. In all cases, the output state can pass the s_a filter, so the ECRP in Definition 8 is satisfied.

(2) In the case that the last round of the full syndrome measurement has some faults, the outcome bundle of the last round may not correspond to the data error before the error correction in step 3. Fortunately, since the outcome bundles are repeated $t+1$ times in a row and there are no more than t faults during the whole protocol, we know that at least one round in the last $t+1$ rounds must be correct, and the outcome bundle of the last round must correspond to the data error right before the last correct round. Let E_{in} be the input error, E_a be the combined error arising from s_a faults that happen before the last correct round, and let E_b be the combined error arising from s_b faults that happen after the last correct round, where the total number of faults is $s = s_a + s_b \leq t$ (see Fig. 14). First, consider the case that E_{in} is in $\mathcal{E}_r$, where $r + s \leq t$. By an analysis similar to that presented in (1), we find that both X and Z parts of $E_a \cdot E_{\text{in}}$ are the combined errors of some fault combinations in $\mathcal{F}_{r+s_a}$, and F_x and F_z from steps 1(a) and 2(a) can completely remove $E_a \cdot E_{\text{in}}$. Thus, the output data error after step 3 is E_b. Since $s_b \leq t$ and the cumulative flag vectors do not change after the last correct round, we find that E_b is the combined error of some fault combination

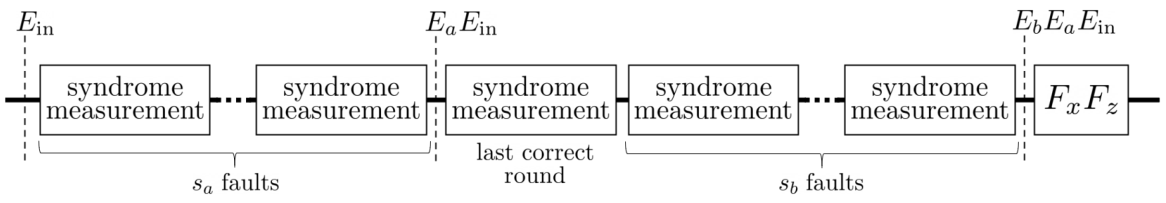


FIG. 14. Fault-tolerant error-correction protocol for a capped color code.

arising from s_b faults whose cumulative flag vector is zero; that is, E_b is in $\mathcal{E}_{s_b}$, where $s_b \leq t$. For this reason, E_b can pass the s filter and can be corrected by the ideal decoder, meaning that both ECCP and ECRP in Definition 8 are satisfied. In contrast, if $E_{\rm in}$ is not in $\mathcal{E}_r$, where $r+s\leq t$, $E_{\rm in}$ may not correspond to any fault combination in $\mathcal{F}_t$, and F_x or F_z may be picked as in step 1(b) or 2(b). Similar to the previous analysis, $F_x \cdot F_z$ will have the same syndrome as that of $E_a \cdot E_{\rm in}$. By an operation in step 3, the output state will be a valid codeword with error E_b, which can pass the s filter. Therefore, the ECRP in Definition 8 is satisfied in this case.

In addition to the capped color code in H form, the FTEC protocol above is also applicable to any CSS code in which $\mathcal{F}_t$ is distinguishable and the possible X-type and Z-type errors are of the same form [i.e., a code to which Definition 15 is applicable for all $r \in \{1,\ldots,t\}$, $t \leq \lfloor (d-1)/2 \rfloor$]. Besides this, we can also construct a FTEC protocol for a general stabilizer code whose circuits for the syndrome measurement give a distinguishable fault set (a code in which $\mathcal{E}_r$ is defined by Definition 11 instead of Definition 15) using similar ideas. A FTEC protocol for such a code is provided in Appendix B.

Because a recursive capped color code in H form of distance d is constructed by recursively encoding the top qubit of the capped color code in H form of distance d using capped color codes of smaller distances, a FTEC protocol for a recursive capped color code in H form can be constructed similarly to a FTEC protocol for a concatenated code. The FTEC protocol is as follows.

FTEC protocol for a recursive capped color code in H form. For each $j=3,5,7,\ldots,d$, perform error correction on the first j layers of the recursive capped color code of distance d using the FTEC protocol for a capped color code in H form of distance j.

C. Fault-tolerant measurement and state preparation protocols

Besides FTEC protocols, we also need other gadgets such as FTM, FTP, and FTG gadgets in order to perform fault-tolerant quantum computation. Note that the definitions of the r filter (Definition 12) and the ideal decoder (Definition 13) depend on how the distinguishable error set is defined. Therefore, in order to utilize the new definitions of fault-tolerant gadgets in Definition 14, all protocols used in the computation must share the same definition of a distinguishable error set. In this section, we construct a FTM protocol for a capped color code in H form, which is also applicable to other CSS codes with similar properties. The distinguishable error set being used in the construction of the FTM protocol will be similar to the distinguishable error set defined for the FTEC protocol for the same code. In addition, a FTP protocol can also be obtained from the FTM protocol.

We start by constructing a FTM protocol for a capped color code in H form obtained from CCC(d). The FTM protocol discussed below can be used to fault-tolerantly measure any logical X or logical Z operator of the form $X^{\otimes n}M$ or $Z^{\otimes n}N$, where M,N are some stabilizers. Let L be the logical operator being measured. We assume that the circuits for measuring X-type and Z-type generators are similar to those used in the FTEC protocol for a capped color code, which give a distinguishable fault set $\mathcal{F}_t$ with $t=(d-1)/2$ (the list of possible fault combinations for the FTM protocol is the same as the list used in the FTEC protocol). In addition, we can always use a nonflag circuit with an arbitrary gate ordering for measuring L (since any error arising from the circuit faults can always be corrected, as we will see later in the protocol analysis). For the FTM protocol, the outcome bundle will be defined as $(m,\vec{\mathbf{s}},\vec{\mathbf{f}})$, where m is the measurement outcome of the logical operator L ($m=0$ and $m=1$ correspond to the $+1$ and -1 eigenvalues of L), and $\vec{\mathbf{s}}=(\vec{\mathbf{s}}_x|\vec{\mathbf{s}}_z)$ and $\vec{\mathbf{f}}=(\vec{\mathbf{f}}_x|\vec{\mathbf{f}}_z)$ are the syndrome and the cumulative flag vector obtained from the measurements of X-type and Z-type generators ($\vec{\mathbf{f}}$ is accumulated from the first round until the current round). An FTM protocol is as follows.

FTM protocol for a capped color code in H form. During a single round of logical operator and full syndrome measurements, measure the operators in the following order: measure L, then the v_i^x, then the $f_i^{\,x}$, then the v_i^z, then the $f_i^{\,z}$. Perform logical operator and full syndrome measurements until the outcome bundles $(m,\vec{\mathbf{s}},\vec{\mathbf{f}})$ are repeated $t+1$ times in a row. Afterwards, do the following.

1. Determine an EC operator F_x using the list of possible fault combinations as follows.

   (a) If there is a fault combination on the list whose syndrome and cumulative flag vector are $(\vec{0}|\vec{\mathbf{s}}_z)$

and $(\vec{\mathbf{f}}_x|\vec{0})$, then F_x is the combined error of such a fault combination. [If there is more than one fault combination corresponding to $(\vec{0}|\vec{\mathbf{s}}_z)$ and $(\vec{\mathbf{f}}_x|\vec{0})$, a combined error of any such fault combinations will work.]

(b) If none of the fault combinations on the list corresponds to $(\vec{0}|\vec{\mathbf{s}}_z)$ and $(\vec{\mathbf{f}}_x|\vec{0})$, then F_x can be any Pauli-X operator whose syndrome is $(\vec{0}|\vec{\mathbf{s}}_z)$.

2. Determine an EC operator F_z using the list of possible fault combinations as follows.

(a) If there is a fault combination on the list whose syndrome and cumulative flag vector are $(\vec{\mathbf{s}}_x|\vec{0})$ and $(\vec{0}|\vec{\mathbf{f}}_z)$, then F_z is the combined error of such a fault combination. [If there is more than one fault combination corresponding to $(\vec{\mathbf{s}}_x|\vec{0})$ and $(\vec{0}|\vec{\mathbf{f}}_z)$, a combined error of any such fault combinations will work.]

(b) If none of the fault combinations on the list corresponds to $(\vec{\mathbf{s}}_x|\vec{0})$ and $(\vec{0}|\vec{\mathbf{f}}_z)$, then F_z can be any Pauli-Z operator whose syndrome is $(\vec{\mathbf{s}}_x|\vec{0})$.

3. Apply $F_x \cdot F_z$ to the data qubits to perform error correction.
4. If L and $F_x \cdot F_z$ anticommute, modify m from 0 to 1 or from 1 to 0. If L and $F_x \cdot F_z$ commute, do nothing.
5. Output m is the operator measurement outcome, where $m = 0$ and $m = 1$ correspond to the $+1$ and -1 eigenvalues of L. If L is a logical Z operator, the output state is the logical $|0\rangle$ or logical $|1\rangle$ state for $m = 0$ or 1. If L is a logical X operator, the output state is the logical $|+\rangle$ or logical $|-\rangle$ state for $m = 0$ or 1.

To verify that the FTM protocol for a capped color code is fault tolerant according to the revised definition (Definition 14), we show that both of the properties in Definition 10 are satisfied when the r filter, the ideal decoder, and the distinguishable error set $\mathcal{E}_r$ are defined as in Definitions 12, 13, and 15. The distinguishable fault set $\mathcal{F}_t$ for this protocol is the same fault set as that defined for the FTEC protocol (i.e., $\mathcal{F}_t$ concerns the circuits for measuring X-type and Z-type generators, and does not concern the circuit for measuring L). We also assume that there are no more than t faults during the whole protocol, so the outcome bundles must be repeated $t+1$ times in a row within $(t+1)^2$ rounds. First, suppose that the operator being measured, L, is a logical Z operator. The analysis will be divided into two cases: (1) the case that the last round of operator and full syndrome measurements has no faults, and (2) the case that the last round of operator and full syndrome measurements has some faults.

(1) Because the last round is correct and the outcome bundles are repeated $(t+1)$ times in a row, $m, \vec{\mathbf{s}}$, and $\vec{\mathbf{f}}$ exactly correspond to the error on the state before step 3. Let $E_{\text{in}} \in \mathcal{E}_r$ be the input error, E_a be the combined error arising from s_a faults in the circuits for measuring L, and let E_b be the combined error arising from s_b faults in the syndrome measurement circuits, where $r + s_a + s_b \leq t$. Also, assume that the (uncorrupted) input state is $|\bar{m}_{\text{in}}\rangle$, where $m_{\text{in}} = 0$ or 1. The data error on the state before the last round is $E_bE_aE_{\text{in}}$. Since L is of the form $Z^{\otimes n}N$, where N is some stabilizer, the X part of E_a has weight no more than s_a, while the Z part of E_a can be any Z-type error. We find that the X part of $E_bE_aE_{\text{in}}$, denoted as $(E_bE_aE_{\text{in}})_x$, is similar to a combined error of X type of some fault combination in $\mathcal{F}_{r+s_a+s_b}$. However, the Z part of $E_bE_aE_{\text{in}}$, denoted as $(E_bE_aE_{\text{in}})_z$, may or may not correspond to a Z-type error of some fault combination in $\mathcal{F}_t$. By picking F_x and F_z as in steps 1 and 2, F_x is logically equivalent to $(E_bE_aE_{\text{in}})_x$ and F_z is logically equivalent to $(E_bE_aE_{\text{in}})_z$ or $(E_bE_aE_{\text{in}})_zZ^{\otimes n}$. So after the error correction in step 3, the output state is $|\bar{m}_{\text{in}}\rangle$ or $Z^{\otimes n}|\bar{m}_{\text{in}}\rangle$. Note that $|\bar{m}_{\text{in}}\rangle$ and $Z^{\otimes n}|\bar{m}_{\text{in}}\rangle$ are the same state for both $m_{\text{in}} = 0$ and $m_{\text{in}} = 1$ cases (the -1 global phase can be neglected in the case of $m_{\text{in}} = 1$).

Next, let us consider the result m obtained from the last round, which tells us whether the state before the measurement of L during the last round is the $+1$ or -1 eigenstate of L. We find that if $m_{\text{in}} = 0$, $m = 0$ whenever $E_bE_aE_{\text{in}}$ commutes with L, and $m = 1$ whenever $E_bE_aE_{\text{in}}$ anticommutes with L. On the other hand, if $m_{\text{in}} = 1$, $m = 1$ whenever $E_bE_aE_{\text{in}}$ commutes with L, and $m = 0$ whenever $E_bE_aE_{\text{in}}$ anticommutes with L. Also, note that $F_x \cdot F_z$ is either $E_bE_aE_{\text{in}}$ or $E_bE_aE_{\text{in}}Z^{\otimes n}$ and L is a logical Z operator, so $E_bE_aE_{\text{in}}$ commutes (or anticommutes) with L if and only if $F_x \cdot F_z$ commutes (or anticommutes) with L. Thus, we need to flip the output as in step 4 whenever $F_x \cdot F_z$ anticommutes with L so that $m = m_{\text{in}}$. As a result, the measurement protocol gives an output state $|\bar{m}_{\text{in}}\rangle$ and its corresponding measurement outcome $m = m_{\text{in}}$ that reflect the uncorrupted input state.

Now, let us consider the case that the uncorrupted input state is of the form $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$. If there is at least one round before the last correct round in which the measurement of L is correct, then the superposition state collapses and the state before the last correct round is either $E_bE_aE_{\text{in}}|\bar{0}\rangle$ or $E_bE_aE_{\text{in}}|\bar{1}\rangle$, so the analysis above is applicable. However, if the measurements of L before the last correct round are all incorrect, it is possible that the superposition state may not collapse and the state before the last correct round is of the form $E_bE_aE_{\text{in}}(\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle)$. Suppose that the measurement of L in the last correct round gives $m = 0$. Then the output state from the last correct round is a $+1$ eigenstate of L, which is $E_bE_aE_{\text{in}}|\bar{0}\rangle$ if $E_bE_aE_{\text{in}}$ commutes with L, or $E_bE_aE_{\text{in}}|\bar{1}\rangle$ if $E_bE_aE_{\text{in}}$ anticommutes with L. In contrast, if the measurement of L in the last correct round gives $m = 1$, then the output state from the last correct round is a -1 eigenstate of L.

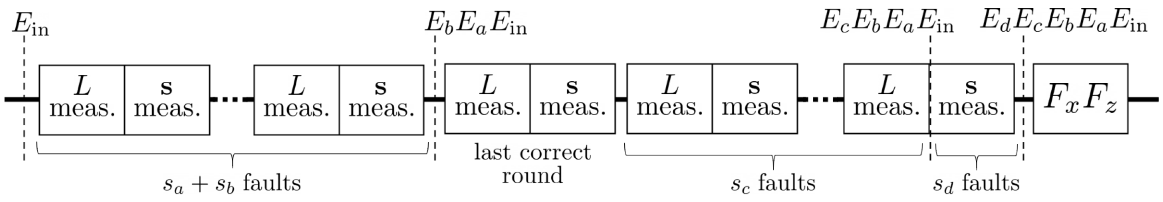


FIG. 15. Fault-tolerant measurement protocol for a capped color code.

This state is $E_bE_aE_{\text{in}}|\bar{1}\rangle$ if $E_bE_aE_{\text{in}}$ commutes with L, or $E_bE_aE_{\text{in}}|\bar{0}\rangle$ if $E_bE_aE_{\text{in}}$ anticommutes with L. By applying $F_x \cdot F_z$ as in step 3 and modifying m whenever $F_x \cdot F_z$ anticommutes with L as in step 4, the outputs from the protocol are either $m = 0$ and $|\bar{0}\rangle$, or $m = 1$ and $|\bar{1}\rangle$ (up to some global phase). Therefore, MCP and MPP in Definition 10 are both satisfied.

(2) In the case that the last round has some faults, because the outcome bundles are repeated $(t + 1)$ times in a row and there are no more than t faults in the protocol, there must be at least one correct round in the last $t + 1$ rounds, and the outcome bundles correspond to the error on the state before the last correct round. Let $E_{\text{in}} \in \mathcal{E}_r$ be the input error, E_a be the combined error arising from s_a faults in the circuits for measuring L before the last correct round, E_b be the combined error arising from s_b faults in the syndrome measurement circuits before the last correct round, E_c be the combined error arising from s_c faults in any circuits after the last correct round but before the syndrome measurement circuits of the very last round, and let E_d be the combined error arising from s_d faults in the syndrome measurement circuits of the very last round, where $r + s_a + s_b + s_c + s_d \leq t$ (see Fig. 15). By an analysis similar to (1), we find that F_x from step 1 is logically equivalent to $(E_bE_aE_{\text{in}})_x$, and F_z from step 2 is logically equivalent to $(E_bE_aE_{\text{in}})_z$ or $(E_bE_aE_{\text{in}})_zZ^{\otimes n}$.

Now, let us consider E_c, which can arise from the circuits for measuring L or the syndrome measurement circuits, and E_d, which can arise from the syndrome measurement circuits. Because the syndromes and the cumulative flag vectors do not change after the last correct round, and because i faults in the circuits for measuring L cannot cause an X-type error of weight more than i, the X part of E_c [denoted as $(E_c)_x$] is similar to the combined error of X type of a fault combination arising from s_c faults whose cumulative flag vector is zero, i.e., $(E_c)_x$ is an error in $\mathcal{E}^x_{s_c}$. In contrast, because the circuits for measuring L can cause a Z-type error of any weight but the syndromes and the cumulative flag vectors do not change after the last correct round, the Z part of E_c [denoted as $(E_c)_z$] can be written as $(\tilde{E}_c)_z$ or $(\tilde{E}_c)_zZ^{\otimes n}$, where $(\tilde{E}_c)_z \in \mathcal{E}^z_{s_c}$. That is, E_c is either $\tilde{E}_c$ or $\tilde{E}_cZ^{\otimes n}$, where $\tilde{E}_c \in \mathcal{E}_{s_c}$. For E_d, which arises from s_d faults in the syndrome measurement circuits in the very last round, we find that it is an error in $\mathcal{E}_{s_d}$ since the cumulative flag vector from the very last round remains the same.

Let the (uncorrupted) input state be of the form $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$. Suppose that the measurement outcome of L from the last correct round is $m = 0$. From the argument on a superposition state in (1), we find that the output state from the last correct round is $E_bE_aE_{\text{in}}|\bar{0}\rangle$ if $E_bE_aE_{\text{in}}$ commutes with L, or $E_bE_aE_{\text{in}}|\bar{1}\rangle$ if $E_bE_aE_{\text{in}}$ anticommutes with L. Thus, the state before step 3 is $E_d\tilde{E}_cE_bE_aE_{\text{in}}|\bar{0}\rangle$ or $E_d\tilde{E}_cZ^{\otimes n}E_bE_aE_{\text{in}}|\bar{0}\rangle$ if $E_bE_aE_{\text{in}}$ commutes with L, or $E_d\tilde{E}_cE_bE_aE_{\text{in}}|\bar{1}\rangle$ or $E_d\tilde{E}_cZ^{\otimes n}E_bE_aE_{\text{in}}|\bar{1}\rangle$ if $E_bE_aE_{\text{in}}$ anticommutes with L. Recall that $F_x \cdot F_z$ is either $E_bE_aE_{\text{in}}$ or $E_bE_aE_{\text{in}}Z^{\otimes n}$, and that $E_bE_aE_{\text{in}}$ commutes (or anticommutes) with L if and only if $F_x \cdot F_z$ commutes (or anticommutes) with L. By applying $F_x \cdot F_z$ as in step 3 and modifying m whenever $F_x \cdot F_z$ anticommutes with L as in step 4, the protocol either outputs $m = 0$ with the output state $E_d\tilde{E}_c|\bar{0}\rangle$ (up to some global phase), or outputs $m = 1$ with the output state $E_d\tilde{E}_c|\bar{1}\rangle$ (up to some global phase). Similar results will be obtained in the case that the measurement outcome of L from the last correct round is $m = 1$.

Because (1) $E_d\tilde{E}_c \in \mathcal{E}_s$, where $s = s_a + s_b + s_c + s_d$ and $r + s \leq t$, (2) the output bit corresponds to the logical qubit of the output state in every case, and (3) the output bit is 0 (or 1) if the (uncorrupted) input state is $|\bar{0}\rangle$ (or $|\bar{1}\rangle$), we find that both of MCP and MPP in Definition 10 are satisfied. A similar analysis can be made for the case that L is a logical X operator. In that case, we let $m = 0$ and $m = 1$ correspond to $|\bar{+}\rangle$ and $|\bar{-}\rangle$, and the analysis similar to (1) and (2) can be applied.

In addition, it is possible to construct a FTP protocol from the FTM protocol described above. For example, if we want to prepare the state $|\bar{0}\rangle$, we can do so by applying the FTM protocol for a logical Z operator to any state, and then applying a logical X operator on the output state if $m = 1$ or do nothing if $m = 0$.

The FTM and the FTP protocols presented in this section are also applicable to any CSS code in which the number of encoded qubits is 1, $\mathcal{F}_t$ is distinguishable (where $\mathcal{F}_t$ corresponds to the circuits for measuring code generators), and the errors in $\mathcal{E}^x_r$ and $\mathcal{E}^z_r$ have the same form for all $r = 1, \ldots, t$, $t \leq \lfloor (d-1)/2 \rfloor$.

Similar to the FTEC protocol for a recursive capped color code, we can construct a FTM protocol for a recursive capped color code similarly to a FTM protocol for a concatenated code. The FTM protocol is as follows.

FTM protocol for a recursive capped color code in H form. Let $L^{(j)}$ be a logical Z (or logical X) operator of a recursive capped color code of distance j. The following procedure can fault-tolerantly measure $L^{(d)}$ on a recursive capped color code of distance d: for each $j = 3, 5, 7, \ldots, d$, perform $L^{(j)}$ measurement on the first j layers of the recursive capped color code of distance d using the FTM protocol for a capped color code in H form of distance j.

A FTP protocol for a recursive capped color code is similar to the FTM protocol for a recursive capped color code, except that some logical operator will be applied to the output state depending on the measurement outcome so that the desired logical state can be obtained.

D. Transversal Clifford gates

From the properties of a capped color code in H form discussed in Sec. IV A, we know that H, S, and CNOT gates are transversal. These gates can play an important role in fault-tolerant quantum computation because transversal gates satisfy both properties of fault-tolerant gate gadgets originally proposed in Ref. [10] (Definition 7). However, since the definition of fault-tolerant gadgets being used in this work is revised as in Definition 14, transversal gates that satisfy the old definition may or may not satisfy the new one. In this section, we show that transversal H, S, and CNOT gates are still fault tolerant according to the new definition of fault-tolerant gadgets when the distinguishable error set $\mathcal{E}_r$ of a capped (or a recursive capped) color code in H form is defined as in Definition 15.

We start by observing the operations of H, S, and CNOT gates. These gates can transform Pauli operators as follows:

$$\begin{array}{rlll} H: & X \mapsto Z, & Y \mapsto -Y, & Z \mapsto X, \\ S: & X \mapsto Y, & Y \mapsto -X, & Z \mapsto Z, \\ \text{CNOT}: & XI \mapsto XX, & ZI \mapsto ZI, & \\ & IX \mapsto IX, & IZ \mapsto ZZ. & \end{array}$$

Meanwhile, the transversal H, S, and CNOT gates can map logical operators $\bar{X} = X^{\otimes n}$ and $\bar{Z} = Z^{\otimes n}$ as follows:

$$\begin{array}{rll} H^{\otimes n}: & \bar{X} \mapsto \bar{Z}, & \bar{Z} \mapsto \bar{X}, \\ S^{\otimes n}: & \bar{X} \mapsto -\bar{Y}, & \bar{Z} \mapsto \bar{Z}, \\ \text{CNOT}^{\otimes n}: & \bar{X} \otimes \bar{I} \mapsto \bar{X} \otimes \bar{X}, & \bar{Z} \otimes \bar{I} \mapsto \bar{Z} \otimes \bar{I}, \\ & \bar{I} \otimes \bar{X} \mapsto \bar{I} \otimes \bar{X}, & \bar{I} \otimes \bar{Z} \mapsto \bar{Z} \otimes \bar{Z}. \end{array}$$

Here $\bar{I} = I^{\otimes n}$, $\bar{Y} = i\bar{X}\bar{Z} = -Y^{\otimes n}$, and $n = 3(d^2 + 1)/2$ is the total number of qubits for each CCC(d) [since $d = 3, 5, 7, \ldots$, we find that $n = 3 \pmod 4$ and $\bar{Y} = -Y^{\otimes n}$ for any CCC(d)]. In addition, the coding subspace is preserved under the $H^{\otimes n}$, $S^{\otimes n}$, or CNOT$^{\otimes n}$ operation (i.e., each stabilizer is mapped to another stabilizer). Therefore, $H^{\otimes n}$, $S^{\otimes n}$, and CNOT$^{\otimes n}$ are logical H, logical $S^\dagger$, and logical CNOT gates, respectively.

For an [[$n, 1, d$]] recursive capped color code in H form in which $n = (d^3 + 3d^2 + 3d - 3)/4$, we find that $n = 3 \pmod 4$ when $d = 3, 7, 11, \ldots$, and $n = 1 \pmod 4$ when $d = 5, 9, 13, \ldots$. That is, $S^{\otimes n}$ is a logical $S^\dagger$ gate when $d = 3, 7, 11, \ldots$, and $S^{\otimes n}$ is a logical S gate when $d = 5, 9, 13, \ldots$. For a recursive capped color code in H form of any distance, $H^{\otimes n}$ and CNOT$^{\otimes n}$ are logical H and logical CNOT gates.

Next, we verify whether the new definition of fault-tolerant gate gadgets in Definition 11 is satisfied. We start by considering logical H and CNOT gates. Let the distinguishable error set $\mathcal{E}_r$ ($r = 1, \ldots, t$) be defined as in Definition 15, where the distinguishable fault set $\mathcal{F}_t$ is the same fault set as that defined for the FTEC protocol for a capped color code in H form. Suppose that the $H^{\otimes n}$ or CNOT$^{\otimes n}$ operation has s faults, the input error of the $H^{\otimes n}$ operation is an error in $\mathcal{E}_r$, where $r + s \leq t$, and that the input error of the CNOT$^{\otimes n}$ operation is an error in $\mathcal{E}_{r_1} \times \mathcal{E}_{r_2}$, where $r_1 + r_2 + s \leq t$. The input error for $H^{\otimes n}$ can be written as $E_1^x \cdot E_2^z$, where $E_1^x \in \mathcal{E}_r^x$ and $E_2^z \in \mathcal{E}_r^z$, and the input error for the CNOT$^{\otimes n}$ operation can be written as $(E_3^x \otimes E_4^x) \cdot (E_5^z \otimes E_6^z)$, where $E_3^x \in \mathcal{E}_{r_1}^x$, $E_4^x \in \mathcal{E}_{r_2}^x$, $E_5^z \in \mathcal{E}_{r_1}^z$, $E_6^z \in \mathcal{E}_{r_2}^z$. Let E_i^x and E_i^z be X-type and Z-type operators that act on the same qubits. We find that

1. $H^{\otimes n}$ maps $E_1^x \cdot E_2^z$ to $E_1^z \cdot E_2^x$, which is an error in $\mathcal{E}_r$;
2. CNOT$^{\otimes n}$ maps $(E_3^x \otimes E_4^x) \cdot (E_5^z \otimes E_6^z)$ to $(E_3^x \otimes E_3^x E_4^x) \cdot (E_5^z E_6^z \otimes E_6^z)$, which is an error in $\mathcal{E}_{r_1+r_2} \times \mathcal{E}_{r_1+r_2}$.

The operation of a logical S gate can be tricky to analyze since it can map X-type errors to a product of X- and Z-type errors (up to some phase factor). Let us consider a single-qubit error $P \in \{I, X, Y, Z\}$, an error from a single CNOT fault during the measurement of an X-type generator that is of the form $P \otimes X^{\otimes m}$, and an error from a single CNOT fault during the measurement of a Z-type generator that is of the form $P \otimes Z^{\otimes m}$ (where $m \geq 0$). The operation of $S^{\otimes n}$ will transform such errors as follows (up to some phase factor):

$$\begin{array}{rcl} I \otimes X^{\otimes m} & \mapsto & (I \otimes X^{\otimes m}) \cdot (I \otimes Z^{\otimes m}), \\ X \otimes X^{\otimes m} & \mapsto & (X \otimes X^{\otimes m}) \cdot (Z \otimes Z^{\otimes m}), \\ Y \otimes X^{\otimes m} & \mapsto & (X \otimes X^{\otimes m}) \cdot (I \otimes Z^{\otimes m}), \\ Z \otimes X^{\otimes m} & \mapsto & (I \otimes X^{\otimes m}) \cdot (Z \otimes Z^{\otimes m}), \\ I \otimes Z^{\otimes m} & \mapsto & (I \otimes I^{\otimes m}) \cdot (I \otimes Z^{\otimes m}), \\ X \otimes Z^{\otimes m} & \mapsto & (X \otimes I^{\otimes m}) \cdot (Z \otimes Z^{\otimes m}), \\ Y \otimes Z^{\otimes m} & \mapsto & (X \otimes I^{\otimes m}) \cdot (I \otimes Z^{\otimes m}), \\ Z \otimes Z^{\otimes m} & \mapsto & (I \otimes I^{\otimes m}) \cdot (Z \otimes Z^{\otimes m}). \end{array}$$

We can see that any error from a single fault will be transformed to an error of the form $E_x \cdot E_z$, where E_x and E_z are X- and Z-type errors from a single fault. For this reason, a combined error from r faults, $\mathbf{E} = E_1 \cdots E_r$, will be transformed to $(\bar{S}E_1\bar{S}^\dagger)\cdots(\bar{S}E_r\bar{S}^\dagger)$, which is of the form $\mathbf{E}_x \cdot \mathbf{E}_z$, where $\mathbf{E}_x$ and $\mathbf{E}_z$ are X- and Z-type errors from r faults. That is, the error after the transformation of $S^{\otimes n}$ is an error in $\mathcal{E}_r$.

In addition, s faults during the application of the $H^{\otimes n}$ or $S^{\otimes n}$ operation can cause an error in $\mathcal{E}_s$, and s faults during the application of the CNOT$^{\otimes n}$ operation can cause an error in $\mathcal{E}_s \times \mathcal{E}_s$. Combining the input error and the error from faults, we find that an output error from the $H^{\otimes n}$ or $S^{\otimes n}$ operation is an error in $\mathcal{E}_{r+s}$, while an output error from the CNOT$^{\otimes n}$ operation is an error in $\mathcal{E}_{r_1+r_2+s} \times \mathcal{E}_{r_1+r_2+s}$. As a result, the $H^{\otimes n}$, $S^{\otimes n}$, and CNOT$^{\otimes n}$ operation satisfy GCP and GPP in Definition 7 when the r filter, the ideal decoder, and the distinguishable error set are defined in Definitions 12, 13, and 15. That is, transversal H, S, and CNOT gates are fault tolerant according to the revised definition. Similar analysis is also applicable to a recursive capped color code in H form. Since the Clifford group can be generated by H, S, and CNOT gates [63,64], any Clifford gate can be fault-tolerantly implemented on a capped (or a recursive capped) color code in H form using transversal H, S, and CNOT gates.

Note that whether a transversal gate satisfies the revised definition of fault-tolerant gate gadgets in Definition 14 depends on how the distinguishable error set is defined (as in either Definition 11 or 15). For example, if the input error E_{in} can arise from t faults (E_{in} is in $\mathcal{E}_t$) and a transversal gate transforms such an error to another error E_{out} that cannot arise from $\leq t$ faults (E_{out} is not in $\mathcal{E}_t$), then this transversal gate is not considered fault tolerant.

E. Fault-tolerant implementation of a logical *T* gate via code switching

In order to achieve a universal set of quantum gates, we also need a fault-tolerant implementation of some gate outside the Clifford group [69]. One possible way to implement a non-Clifford gate on the capped color code in H form is to use magic state distillation [70], but large overhead might be required [55]. Another possible way is to perform code switching; since the code in H form possesses transversal H, S, and CNOT gates, and the code in T form possesses a transversal T gate, we can apply transversal H, S, or CNOT gates and perform FTEC on the code in H form, and switch to code in T form to apply a transversal T gate when necessary. However, logical T-gate implementation via code switching on a capped color code might not be fault tolerant since the code in T form constructed from CCC(d) has distance 3 regardless of the parameter d, and a few faults that occurred to the code in T form can cause a logical error. Fortunately, for a recursive capped color code, both distances of the code in H form and the code in T form constructed from RCCC(d) are d. Thus, fault-tolerant T-gate implementation via code switching

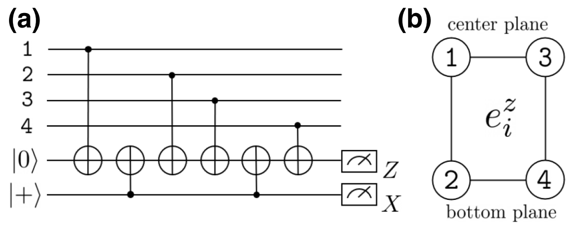


FIG. 16. (a) A flag circuit for measuring a vertical face generator e_i^z. (b) The ordering of data CNOT gates in the circuit for each e_i^z.

is possible. The fault-tolerant protocol for logical T-gate implementation on a recursive capped color code will be developed in this section.

First, let us assume that the T-gate implementation protocol is performed after the FTEC protocol for a recursive capped color code in H form developed in Sec. V B, using the following CNOT orderings.

1. In the preceding FTEC protocol, the `f`, `v`, and `cap` operators on the $(j-2)$th, $(j-1)$th, and jth layers of the recursive capped color code are measured using the CNOT orderings for the `f`, `v`, `cap` operators of a capped color code in H form of distance j ($j = 3, 5, \ldots, d$) that give a distinguishable fault set [where an operator on $\mathtt{q}_0$ of a capped color code is replaced by operators on all qubits on the $(j-2)$th layer of a recursive capped color code].
2. During the switching from the code in H form to T form, all Z-type vertical face generators e_i^z are measured using flag circuits with one flag ancilla similar to the circuit in Fig. 16 (see the definition of vertical face generators in Secs. IV A and IV B).
3. During the switching from the code in T form to H form, all X-type generators of 2D color codes on layers $2, 4, \ldots, d-1$ of the code are measured using circuits similar to those being used in the preceding FTEC protocol.

The logical T-gate implementation protocol will use the following ideas: we start from the recursive capped color code in H form, switch to the code in T form, apply a transversal T gate, switch back to the code in H form, and then perform error correction using a FTEC protocol similar to the FTEC protocol for a recursive capped color code in H form, except that possible faults from e_i^z measurements are also included in the distinguishable fault set (note that we never perform error correction on the code in T form). The full procedure of the T-gate implementation protocol is as follows.

Fault-tolerant *T*-gate implementation protocol for a recursive capped color code in H form.

1. During a single round of operator measurements, measure all Z-type vertical face generators. Perform

measurements until the outcomes are repeated $t+1$ times in a row. After that, apply a Pauli operator corresponding to the repeated measurement outcome (see also the code switching procedure in Secs. IV A and IV B).
2. Perform a logical T operation by applying physical T and $T^\dagger$ gates on qubits represented by black and white vertices, respectively (see also Proposition 7).
3. During a single round of operator measurements, measure all X-type generators of 2D color codes on layers $2,4,\dots,d-1$ of the code. Perform measurements until the outcomes are repeated $t+1$ times in a row. After that, apply a Pauli operator corresponding to the repeated measurement outcome (see also the code switching procedure in Secs. IV A and IV B).
4. Perform error correction using a FTEC protocol similar to the FTEC protocol for a recursive capped color code in H form described in Sec. V B, except that possible faults from vertical face generator measurements are also included in the distinguishable fault set.

We can show that the protocol described above is fault tolerant using the following facts.

1. Both codes in H form and T form have distance d [in fact, the distance of RCCC(d) does not depend on the gauge choice].
2. An input error to the logical T-gate implementation protocol is an error E_{in} in the distinguishable set $\mathcal{E}_r$, where r is the number of faults in the preceding FTEC protocol.
3. During the switching from the code in H form to the code in T form (step 1), the flag outcome is not zero whenever a single fault that leads to a data error of weight 2 occurs. That is, when the flag is zero, s_1 faults will lead to an error E_1 of weight $\leq s_1$.
4. A logical T gate is transversal, so s_2 faults during step 2 will lead to an error E_2 of weight $\leq s_2$.
5. Any fault that can occur during the switching from the code in T form to the code in H form (step 3) will lead to an error on layer 2,4,..., or $d-1$ [a center plane of inner CCC(j), $j=3,5,7,\dots$].
6. The gauge measurements and Pauli operation during the code switching correct the part of the data error that acts on the gauge qubits being measured. The code switching does not affect the part of the data error that acts on the logical qubit.

Consider the data error $E_3E_2\bar{T}E_1E_{\text{in}}\bar{T}^\dagger$ (the total error on the desired state $\bar{T}|\bar{\psi}_{\text{in}}\rangle$). We can show that, when $r+s_1+s_2+s_3\leq t$, $E_3E_2\bar{T}E_1E_{\text{in}}\bar{T}^\dagger$ is correctable by the FTEC protocol in step 4; this is equivalent to showing that $(\bar{T}E'^{\dagger}_{\text{in}}E'^{\dagger}_1\bar{T}^\dagger E'^{\dagger}_2E'^{\dagger}_3)(E_3E_2\bar{T}E_1E_{\text{in}}\bar{T}^\dagger)$ is not a logical operator with zero cumulative flag vector when $r+r'+s_1+s'_1+s_2+s'_2+s_3+s'_3\leq 2t$ (using a technique similar to the proof of Theorem 2). In addition, we know from the analysis of the FTEC protocol in Sec. V B that if the FTEC protocol in step 4 can correct any possible error after step 3 whenever $s_4=0$, then in the case that $s_4\leq t$ the output error will be an error in $\mathcal{E}_{s_4}$.

We point out that the protocol described in this section works for a recursive capped color code in H form of any distance given that *flag circuits* are used in the gauge operator measurements during the code switching. Note that, for the recursive capped color codes in H form of distances 3 and 5, it is possible to obtain a distinguishable fault set when the circuits for generator measurements are *nonflag circuits* (thus, FTEC, FTP, FTM, and fault-tolerant Clifford computation with one ancilla are possible). In that case, however, an additional ancilla is required if one wants to perform logical T-gate implementation via code switching using the fault-tolerant protocol provided in this section.

VI. DISCUSSION AND CONCLUSIONS

In this work, we observe that errors arising from a few faults depend on the structure of the circuits chosen for syndrome measurement, and develop a FTEC protocol accordingly. A fault set that includes all possible fault combinations arising from at most a certain number of faults is said to be distinguishable if any pair of fault combinations in the set either leads to logically equivalent data errors or leads to different syndromes or cumulative flag vectors (as defined in Definition 3). Distinguishability may depend on the number of flag ancillas being used in the circuits, the ordering of gates in the circuits, and the choice of stabilizer generators being measured. If we can find a set of circuits for a stabilizer code that leads to a distinguishable fault set, we can construct a FTEC protocol, as shown in Sec. V B.

We prove in Lemma 1 that if an $[[n,k,d]]$ CSS code has odd n, $k=1$, even weight stabilizer generators, and logical X and Z being $X^{\otimes n}$ and $Z^{\otimes n}$, then two Pauli errors of X type (or Z type) with the same syndrome are logically equivalent if and only if they have the same weight parity. One may note that the weight parity of a Pauli operator and the anticommutation between the Pauli operator and a logical operator are closely related. In fact, for a given stabilizer code, the normalizer group can be generated by the stabilizer generators of the code and all independent logical Pauli operators; for example, the normalizer group of the Steane code is $N(S)=\langle g_i^x,g_i^z,X^{\otimes 7},Z^{\otimes 7}\rangle_{i=1,2,3}$. If the anticommutation between a Pauli error E and each of the generators of $N(S)$ can be found, then a Pauli error logically equivalent to E can be determined with certainty. The EC techniques presented in Ref. [1] and this work use the fact that the weight parity of an error on a smaller code (or the anticommutation between the error and a logical operator of a smaller code) can be inferred by the measurement results of the stabilizer generators of a bigger code. We are hopeful that the relationship between the weight parity and the anticommutation can lead to EC techniques similar to the weight parity technique for a general stabilizer code in which the number of logical qubits can be greater than 1.

With Lemma 1 in mind, we present the 3D color code of distance 3 in Sec. III and construct a family of capped color codes in Sec. IV, which are good candidates for our protocol construction (the 3D color code of distance 3 is the smallest capped color code). A capped color code is a subsystem code; it can be transformed to stabilizer codes, namely capped color codes in H form and T form, by the gauge fixing method. The code in H form has transversal H, S, and CNOT gates, while the code in T form has transversal CNOT and T gates. One interesting property of a capped color code in H form is that the code contains a 2D color code as a subcode lying on the center plane. Since a `cap` generator of X type (or Z type) has support on all qubits on the center plane, the weight parity of an error of Z type (or X type) that occurred on the center plane can be obtained from the measurement result of the `cap` generator. The syndrome of the error on the center plane corresponding to the measurements of the 2D code generators together with the error weight parity can lead to an EC operator for such an error by Lemma 1. Exploiting these facts, we design circuits for measuring generators of a capped color code such that most of the possible errors are on the center plane. We prove in Theorem 1 that if the circuits satisfy some conditions, the fault set corresponding to all possible fault combinations arising from up to $t = (d-1)/2$ faults is distinguishable, where $d = 3, 5, 7, \ldots$ is the distance of the code. Furthermore, we prove in Theorems 2 and 3 that a distinguishable fault set for a capped color code in H form of *any distance* can be obtained, given that circuits for measuring code generators are flag circuits with one flag ancilla of a particular form. We also show that, for the codes of distances 3 and 5, it is possible to obtain a distinguishable fault set using nonflag circuits with specific CNOT orderings. However, whether such nonflag circuits exist for the code of distance 7 or higher is still not known.

Besides capped color codes, we also construct a family of recursive capped color codes in Sec. IV. A recursive capped color code RCCC(d) can be obtained by recursively encoding the top qubit of a capped color code CCC(d) by capped color codes of smaller distances. Similar to a capped color code, stabilizer codes, namely recursive capped color codes in H form and T form, can be obtained by the gauge fixing method. Circuits for measuring code generators that work for capped color codes are also applicable to recursive capped color codes. The main advantage of a recursive capped color code is that both codes in H form and T form have the same distance, allowing us to perform fault-tolerant logical T-gate implementation via code switching.

In Sec. V, we construct several fault-tolerant protocols using the fact that the fault set corresponding to the protocols being used is distinguishable. Our definitions of fault-tolerant gadgets in Definition 14 also use the fact that some errors can be distinguished by their relevant flag information, so they can be viewed as a generalization of the definitions of fault-tolerant gadgets proposed in Ref. [10] (Definitions 7–10). Our protocols are not limited to the capped or the recursive capped color codes; some of the protocols are also applicable to other families of stabilizer codes if their syndrome measurement circuits give a distinguishable fault set. Since possible errors depend on every fault-tolerant gadget being used, all protocols for quantum computation (including error correction, gate, measurement, and state preparation gadgets) must be designed in tandem in order to achieve fault tolerance.

In our development, the ideal decoder and the r filter (which define fault-tolerant gadgets) are defined by a distinguishable error set in which errors correspond to fault combinations with zero cumulative flag vector (see Definitions 11–14). The intuition behind the definitions with zero cumulative flag vector is that in a general flag FTEC protocol we normally repeat the measurements until the outcomes (syndromes and flag vectors) are repeated $t+1$ times in a row. Thus, undetectable faults at the very end of the protocol that give repeated outcomes must correspond to the zero cumulative flag vector (see the analysis of the FTEC protocol in Sec. V B for more details). Note that nontrivial cumulative flag vectors are used to distinguish possible fault combinations arising during the FTEC protocol only; the flag information is used locally in each FTEC gadget and is not passed on to other gadgets. One interesting future direction would be studying how fault-tolerant protocols can be further improved by exploiting the flag information outside of the FTEC protocol. For example, we may define both the ideal decoder and r filter using fault combinations with trivial or nontrivial cumulative flag vectors. However, when a FTEC protocol is allowed to output nontrivial flag information, we have to make sure that other subsequent fault-tolerant gadgets (such as FTG gadgets) must be able to process the flag information in such a way that their possible output errors are still distinguishable. This study is beyond the scope of this work.

One should note that it is possible to use fault-tolerant protocols satisfying the old definitions of fault-tolerant gadgets (Definitions 7–10) in conjunction with fault-tolerant protocols satisfying our definitions of fault-tolerant gadgets (Definition 14). In particular, observe that, for any Pauli error of weight $w \leq t$, we can always find a fault combination arising from w faults whose combined error is such an error; i.e., any Pauli error of weight up to t is contained in a distinguishable fault set $\mathcal{F}_t$. Therefore, a FTEC protocol satisfying Definition 14 can be used to correct an output error of any fault-tolerant protocol satisfying one of the old definitions (assuming that both protocols can tolerate the same number of faults). However, the converse might not be true since a FTEC protocol satisfying the old definition of a FTEC gadget might not be able to correct errors of high weight arising from a small number of faults in the protocol satisfying Definition 14.

In this work, we show that universal quantum computation can be performed fault-tolerantly on a recursive capped color code in H form of any distance. First, we provide FTEC, FTM, and FTP protocols for a capped color code in H form that are applicable to the code of any distance as long as the fault set is distinguishable (see Secs. V B and V C). From the aforementioned protocols, we can construct FTEC, FTM, and FTP protocols for a recursive capped color code in H form similarly to conventional fault-tolerant protocols for a concatenated code. Second, we show that, for a capped color code, transversal H, S, and CNOT gates are fault tolerant according to our revised definitions of fault-tolerant gadgets in Definition 14 (see Sec. V D), and a similar analysis is also applicable to a recursive capped color code. Last, we provide a fault-tolerant protocol for implementing a logical T gate on a recursive capped color code in H form via code switching, which is applicable to the code of any distance given that circuits for measuring gauge operators are flag circuits of a particular form (see Sec. V E).

Compared with other codes with the same distance, capped and recursive capped color codes may not have the fewest number of data qubits. Nevertheless, these codes have some special properties that may be useful for fault-tolerant quantum computation. The numbers of data qubits n (as functions of the code distance d) for the families of 2D color codes [54], capped color codes, recursive capped color codes, traditional 3D color codes [73], and stacked codes [78] are provided in the second column of Table IV. We make the following observations.

1. The number of data qubits required for a capped color code in H form is about twice that of a 2D color code of the same distance [both numbers are $O(d^2)$]. One advantage that capped color codes have over 2D color codes is that a logical T gate can be implemented on the capped color codes via code switching. Although the process might not be fully fault tolerant (because the codes in T form are of distance 3 regardless of d), code switching uses fewer ancillas compared to magic state distillation and may be beneficial if the error rate is low enough.
2. When d is large, the number of data qubits required for a recursive capped color code is about 2 times smaller than that of a 3D color code, and about 3 times smaller than that of a stacked code of the same distance [all numbers are $O(d^3)$]. For these three families of codes, a logical T gate can be fault-tolerantly implemented via code switching since the code distance does not depend on the gauge choice.

Using our fault-tolerant protocols, Clifford computation on a capped color code in H form of any distance can be achieved using only two ancillas, while universal quantum computation on a recursive capped color code in H form of any distance can be achieved using only two ancillas. This is equal to the number of ancillas required for fault-tolerant protocols for Clifford computation on a 2D color code of any distance (by Theorem 2). It should be noted that the aforementioned results on the number of ancillas hold under the assumptions that (1) qubit preparation and qubit measurement are fast enough so that the ancillas can be reused, and that (2.a) all-to-all connectivity between data and ancilla qubits are allowed. In practice, attaining the minimum number of ancillas can be challenging because the qubit connectivity is restricted to the nearest-neighbor interactions in most architectures. A more practical assumption is (2.b) having dedicated syndrome and flag ancillas for each stabilizer generator measurement. For a 2D color code, syndrome and flag ancillas can be shared between X-type and Z-type generators acting on the same set of qubits, so the number of required ancillas is the number of stabilizer generators, which is $3(d^2-1)/4$. For a capped (or recursive capped) color code in H form, syndrome and flag ancillas can be shared between X-type and Z-type volume ($\mathtt{v}$) generators acting on the same set of qubits, and face ($\mathtt{f}$) generators can share ancillas with their corresponding volume generators. Thus, the number of required ancillas is equal to the number of stabilizer generators of the subsystem code CCC(d) [or RCCC(d)]. (Note that, on color codes, generators of the same color can be measured in parallel.)

The total numbers of data and ancilla qubits required for 2D color codes, capped color codes, recursive capped color codes, traditional 3D color codes, and stacked codes under assumptions (1) and (2.b) are displayed in the fourth column of Table IV. (Note that we still do not know the actual minimum numbers of required ancillas for a 3D color code and a stacked code to achieve fault tolerance. The numbers for these two codes in the table are for the case that only one ancilla per generator is required.) We find the following.

1. In exchange for having T-gate implementation via code switching available (although the process is not fully fault tolerant), protocols for a capped color code in H form require about 50% more qubits than those for a 2D color code of the same distance.
2. When $d=5$, the recursive capped color code outperforms the stacked code and is comparable to the 3D color code. When $d\geq 7$, the recursive capped color code outperforms both the 3D color code and stacked code. (These three codes are the same code when $d=3$.)

TABLE IV. Comparison between the numbers of required qubits for a 2D color code, a capped color code (in H form), a recursive capped color code, a traditional 3D color code, and a stacked code of distance d. The assumptions being used in the third and the fourth columns are that (1) qubit preparation and qubit measurement are fast, and that (2.a) all-to-all connectivity between data and ancilla qubits are allowed or (2.b) there are dedicated syndrome and flag ancillas for each generator measurement.

Code family	Number of data qubits only $n(d)$	Number of data and ancilla qubits, assuming (1) and (2.a)	Number of data and ancilla qubits, assuming (1) and (2.b)
2D color code [54]	$3(d^2-1)/4+1$	$3(d^2-1)/4+3$	$3(d^2-1)/2+1$
Capped color code	$3(d^2-1)/2+3$	$3(d^2-1)/2+5$	$9(d^2-1)/4+5$
Recursive capped color code	$(d^3+3d^2+3d-3)/4$	$(d^3+3d^2+3d+5)/4$	$(3d^3+9d^2+13d-17)/8$
3D color code [73]	$(d^3+d)/2$	$(d^3+d+2)/2$[a]	$(7d^3+3d^2+5d-3)/12$[a]
Stacked code [78]	$(3d^3-3d^2+d+3)/4$	$(3d^3-3d^2+d+7)/4$[a]	$(15d^3-15d^2+9d+7)/16$[a]

[a]We still do not know the actual minimum numbers of required ancillas for a 3D color code and a stacked code to achieve fault tolerance. The numbers for these codes in the table are for the case that only one ancilla per generator is required.

Recently, Beverland *et al.* [75] compared the overhead required for T-gate implementation with two methods: using a 2D color code via magic state distillation versus using a (traditional) 3D color code via code switching. They found that magic state distillation outperforms code switching except at some low physical error rate and when certain fault-tolerant schemes are used in the simulation. Since our protocols require only a few ancillas per generator and the data block of a recursive capped color code is smaller than that of a 3D color code of the same distance, we are hopeful that the range of physical error rates in which code switching beats magic state distillation could be improved by our protocols. A careful simulation on the overhead is required, which we leave for future work.

Last, we point out that our fault-tolerant protocols using the flag and the weight parity techniques are specially designed for the *circuit-level noise* so that all possible data errors arising from a few faults (including any one- and two-qubit gate faults, faults during the ancilla preparation and measurement, and faults during wait time) can be corrected. However, our protocols require repeated syndrome measurements in order to avoid syndrome bit flips that may occur during the protocols, and the processes can increase the number of gate operations. Single-shot error correction [84] is one technique that can deal with the syndrome bit flips without using repeated syndrome measurements. We hope that the flag, the weight parity, and the single-shot error-correction techniques could be used together to build fault-tolerant protocols that can protect the data against the circuit-level noise and require only small numbers of gates and ancillas.

ACKNOWLEDGMENTS

We thank Christopher Chamberland and Michael Vasmer for the suggestion of considering weight parity error correction on color codes. We also thank Ken Brown, Robert Calderbank, Arun Aloshious, Rui Chao, Shilin Huang, Eric Sabo, and other members of Duke Quantum Center for helpful discussions on the similarities between capped color codes and stacked codes, the alternative proof of Theorem 2 presented in this work, and possible future directions. We would like to extend our gratitude to Dan Browne, Andrew Cross, David Gosset, Raymond Laflamme, Ted Yoder, and Beni Yoshida for their helpful comments and suggestions. T.T. acknowledges the support of The Queen Sirikit Scholarship under The Royal Patronage of Her Majesty Queen Sirikit of Thailand. D.L. is supported by an NSERC Discovery grant. Perimeter Institute is supported in part by the Government of Canada and the Province of Ontario.

APPENDIX A: PROOF OF THEOREM 1

In the first part of the proof, we assume that data errors arising from all faults are purely Z type, and show that if Conditions 1–5 are satisfied, then there is no fault combination arising from up to $d-1$ faults whose combined error is a logical Z operator and its cumulative flag vector is zero. Because i faults during the measurements of X-type generators cannot cause a Z-type error of weight more than i, we can assume that each fault is either a qubit fault causing a Z-type error (which is a $\mathtt{q}_0$, $\mathtt{q}_{\mathtt{on}}$, or $\mathtt{q}_{\mathtt{off}}$ fault) or a fault during a measurement of some Z-type generator (which is an $\mathtt{f}$, $\mathtt{v}$, $\mathtt{v}^*$, or $\mathtt{cap}$ fault).

First, recall the main equations (in mod 2):

$$s_{\mathtt{cap}} = n_0 + n_{\mathrm{on}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*,\mathtt{cen}}) + \sum \mathrm{WP}(\sigma_{\mathtt{cap}}), \tag{A1}$$

$$\vec{s}_{\mathtt{f}} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{p}_{\mathtt{f}} + \sum \vec{p}_{\mathtt{v}} + \sum \vec{p}_{\mathtt{v}^*,\mathtt{cen}} + \sum \vec{p}_{\mathtt{cap}}, \tag{A2}$$

$$\vec{s}_{\mathrm{v}} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathtt{f}} + \sum \vec{q}_{\mathrm{v}}^{*} + \sum \vec{p}_{\mathtt{cap}}, \quad \text{(A3)}$$

$$\mathrm{WP}_{\mathtt{tot}} = n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}}) + n_{\mathrm{v}}^{*} + \sum \mathrm{WP}(\sigma_{\mathtt{cap}}), \quad \text{(A4)}$$

$$\vec{\mathbf{f}}_{\mathtt{cap}} = \sum \vec{f}_{\mathtt{cap}}, \quad \text{(A5)}$$

$$\vec{\mathbf{f}}_{\mathtt{f}} = \sum \vec{f}_{\mathtt{f}}, \quad \text{(A6)}$$

$$\vec{\mathbf{f}}_{\mathrm{v}} = \sum \vec{f}_{\mathrm{v}} + \sum \vec{f}_{\mathrm{v}}^{*}, \quad \text{(A7)}$$

$$\mathrm{WP}_{\mathtt{bot}} = n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathrm{v}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}^*,\mathtt{bot}}), \quad \text{(A8)}$$

$$\vec{s}_{\mathtt{bot}} = \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathrm{v}} + \sum \vec{p}_{\mathrm{v}^*,\mathtt{bot}}. \quad \text{(A9)}$$

Note that the types of faults involved in the main equations and the types of faults involved in the conditions are related by the correspondences in Table III. Here we show that if Conditions 1–5 are satisfied and there exists a fault combination arising from up to $d-1$ faults that corresponds to a logical Z operator and the zero cumulative flag vector, some contradictions will happen (also note that Condition 0 is automatically satisfied). By Lemma 1, a fault combination corresponding to a logical Z operator and the zero cumulative flag vector gives $s_{\mathtt{cap}} = 0, \vec{s}_{\mathtt{f}} = \vec{0}$, $\vec{s}_{\mathrm{v}} = \vec{0}$, $\mathrm{WP}_{\mathtt{tot}} = 1$, $\vec{\mathbf{f}}_{\mathtt{cap}} = \vec{0}$, $\vec{\mathbf{f}}_{\mathtt{f}} = \vec{0}$, $\vec{\mathbf{f}}_{\mathrm{v}} = \vec{0}$, $\mathrm{WP}_{\mathtt{bot}} = 1$, and $\vec{s}_{\mathtt{bot}} = 0$. We divide the proof into four cases: (1) $n_{\mathtt{f}} = 0$ and $n_{\mathtt{cap}} = 0$, (2) $n_{\mathtt{f}} \geq 1$ and $n_{\mathtt{cap}} = 0$, (3) $n_{\mathtt{f}} = 0$ and $n_{\mathtt{cap}} \geq 1$, and (4) $n_{\mathtt{f}} \geq 1$ and $n_{\mathtt{cap}} \geq 1$.

Case 1: $n_{\mathtt{f}} = 0$ and $n_{\mathtt{cap}} = 0$. The main equations can be simplified as follows (trivial equations are neglected):

$$0 = n_0 + n_{\mathrm{on}} + \sum \mathrm{WP}(\sigma_{\mathrm{v}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}^*,\mathtt{cen}}), \quad \text{(A1)}$$

$$\vec{0} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{p}_{\mathrm{v}} + \sum \vec{p}_{\mathrm{v}^*,\mathtt{cen}}, \quad \text{(A2)}$$

$$\vec{0} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{q}_{\mathtt{off}} + \sum \vec{q}_{\mathrm{v}}^{*}, \quad \text{(A3)}$$

$$1 = n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathrm{v}}^{*}, \quad \text{(A4)}$$

$$\vec{0} = \sum \vec{f}_{\mathrm{v}} + \sum \vec{f}_{\mathrm{v}}^{*}, \quad \text{(A7)}$$

$$1 = n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathrm{v}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}^*,\mathtt{bot}}), \quad \text{(A8)}$$

$$\vec{0} = \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathrm{v}} + \sum \vec{p}_{\mathrm{v}^*,\mathtt{bot}}. \quad \text{(A9)}$$

All faults involved in Eqs. (A3) and (A4) correspond to $\mathtt{q}_{\mathtt{2D}}$ faults on the 2D code and the total number of faults is at most $d-1$. Because Condition 0 is satisfied, from Eqs. (A3) and (A4) we must have $n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathrm{v}}^{*} = 0 \pmod 2$, which implies that $n_0 = 1$. Thus, Eq. (A1) becomes

$$1 = n_{\mathrm{on}} + \sum \mathrm{WP}(\sigma_{\mathrm{v}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}^*,\mathtt{cen}}). \quad \text{(A1)}$$

Since the total number of faults is $n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-1$, we find that $n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-2$. Let us consider the following subcases.

(1.a) If $n_{\mathtt{off}} = 0$, we have $n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-2-n_{\mathrm{on}} \leq d-2$. In this case, Eqs. (A7)–(A9) contradict Condition 1 (where v and v* faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(1.b) If $n_{\mathtt{off}} \geq 1$, we have $n_{\mathrm{on}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-2-n_{\mathtt{off}} \leq d-3$. In this case, Eqs. (A1), (A2), and (A7) contradict Condition 2 (where a $\mathtt{q}_{\mathrm{on}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault, and v and v* faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

Case 2: $n_{\mathtt{f}} \geq 1$ and $n_{\mathtt{cap}} = 0$. The main equations can be simplified as follows:

$$0 = n_0 + n_{\mathrm{on}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}^*,\mathtt{cen}}), \quad \text{(A1)}$$

$$\vec{0} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{p}_{\mathtt{f}} + \sum \vec{p}_{\mathrm{v}} + \sum \vec{p}_{\mathrm{v}^*,\mathtt{cen}}, \quad \text{(A2)}$$

$$\vec{0} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathtt{f}} + \sum \vec{q}_{\mathrm{v}}^{*}, \quad \text{(A3)}$$

$$1 = n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathtt{f}}) + n_{\mathrm{v}}^{*}, \quad \text{(A4)}$$

$$\vec{0} = \sum \vec{f}_{\mathtt{f}}, \quad \text{(A6)}$$

$$\vec{0} = \sum \vec{f}_{\mathrm{v}} + \sum \vec{f}_{\mathrm{v}}^{*}, \quad \text{(A7)}$$

$$1 = n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathrm{v}}) + \sum \mathrm{WP}(\sigma_{\mathrm{v}^*,\mathtt{bot}}), \quad \text{(A8)}$$

$$\vec{0} = \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathrm{v}} + \sum \vec{p}_{\mathrm{v}^*,\mathtt{bot}}. \quad \text{(A9)}$$

The total number of faults is $n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathtt{f}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-1$, which means that $n_{\mathtt{off}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-1-n_0-n_{\mathrm{on}}-n_{\mathtt{f}}$ (where $n_{\mathtt{f}} \geq 1$). Consider the following subcases.

(2.a) If $n_0 = 1$ or $n_{\mathrm{on}} \geq 1$ or $n_{\mathtt{f}} \geq 2$, we have $n_{\mathtt{off}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-3$. In this case, Eqs. (A7)–(A9) contradict Condition 2 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault, and v and v* faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(2.b) If $n_0 = 0, n_{\mathrm{on}} = 0$, and $n_{\mathtt{f}} = 1$, we find that $n_{\mathtt{off}} + n_{\mathtt{f}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-1$ and $n_{\mathtt{off}} + n_{\mathrm{v}} + n_{\mathrm{v}}^{*} \leq d-2$. Let us divide this case into the following further subcases (where some subcases may overlap).

(i) If $n_{\mathrm{v}} \geq 1$, then $n_{\mathtt{off}} + n_{\mathtt{f}} + n_{\mathrm{v}}^{*} \leq d-2$. In this case, Eqs. (A3), (A4), and (A6) contradict Condition 3 (where $\mathtt{q}_{\mathtt{off}}$ and $\mathtt{q}_{\mathtt{v}^*}$ faults correspond to a $\mathtt{q}_{\mathtt{2D}}$ fault, and an $\mathtt{f}$ fault corresponds to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(ii) If $n_{\mathtt{v}} = 0$ and $n_{\mathtt{v}}^* = 0$, then Eqs. (A8) and (A9) contradict Condition 0 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault).

(iii) If $n_{\mathtt{off}} = 0$, then $n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d-2$ and Eqs. (A7)–(A9) contradict Condition 1 (where $\mathtt{v}$ and $\mathtt{v}^*$ faults correspond an to $\mathtt{f}_{\mathtt{2D}}$ fault).

(iv) If $n_{\mathtt{off}} \geq 1$, $n_{\mathtt{v}} = 0$, and $n_{\mathtt{v}}^* = 1$, then $n_{\mathtt{off}} + n_{\mathtt{v}}^* \leq d-2$ and Eqs. (A7)–(A9) contradict Condition 3 (where $\mathtt{q}_{\mathtt{off}}$ fault correspond to a $\mathtt{q}_{\mathtt{2D}}$ fault, and a $\mathtt{v}^*$ fault corresponds to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(v) If $n_{\mathtt{off}} \geq 1$, $n_{\mathtt{v}} = 0$, $n_{\mathtt{v}}^* \geq 2$, and $n_{\mathtt{off}} + n_{\mathtt{f}} + n_{\mathtt{v}}^* \leq d-2$, then Eqs. (A3), (A4) and (A6) contradict Condition 3 (where $\mathtt{q}_{\mathtt{off}}$ and $\mathtt{q}_{\mathtt{v}^*}$ faults correspond to a $\mathtt{q}_{\mathtt{2D}}$ fault, and an $\mathtt{f}$ fault corresponds to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(vi) If $n_{\mathtt{off}} \geq 1$, $n_{\mathtt{v}} = 0$, $n_{\mathtt{v}}^* \geq 2$, and $n_{\mathtt{off}} + n_{\mathtt{f}} + n_{\mathtt{v}}^* = d-1$, then Eqs. (A1),(A2), and (A6)–(A9) contradict Condition 4 (where $\mathtt{q}_{\mathtt{off}}$, $\mathtt{q}_{\mathtt{f}}$, and $\mathtt{q}_{\mathtt{v}^*}$ faults correspond to $\mathtt{q}_{\mathtt{2D}}$, $\mathtt{f}_{\mathtt{2D}}$, and $\mathtt{v}^*_{\mathtt{2D}}$ faults, respectively).

Case 3: $n_{\mathtt{f}} = 0$ and $n_{\mathtt{cap}} \geq 1$. The main equations can be simplified as follows:

$$0 = n_0 + n_{\mathrm{on}} + \sum \mathrm{WP}(\sigma_{\mathtt{v}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*,\mathrm{cen}}) + \sum \mathrm{WP}(\sigma_{\mathtt{cap}}), \tag{A1}$$

$$\vec{0} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{p}_{\mathtt{v}} + \sum \vec{p}_{\mathtt{v}^*,\mathrm{cen}} + \sum \vec{p}_{\mathtt{cap}}, \tag{A2}$$

$$\vec{0} = \sum \vec{q}_{\mathrm{on}} + \sum \vec{q}_{\mathtt{off}} + \sum \vec{q}_{\mathtt{v}}^* + \sum \vec{p}_{\mathtt{cap}}, \tag{A3}$$

$$1 = n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathtt{v}}^* + \sum \mathrm{WP}(\sigma_{\mathtt{cap}}), \tag{A4}$$

$$\vec{0} = \sum \vec{f}_{\mathtt{cap}}, \tag{A5}$$

$$\vec{0} = \sum \vec{f}_{\mathtt{v}} + \sum \vec{f}_{\mathtt{v}}^*, \tag{A7}$$

$$1 = n_{\mathtt{off}} + \sum \mathrm{WP}(\sigma_{\mathtt{v}}) + \sum \mathrm{WP}(\sigma_{\mathtt{v}^*,\mathrm{bot}}), \tag{A8}$$

$$\vec{0} = \sum \vec{q}_{\mathtt{off}} + \sum \vec{p}_{\mathtt{v}} + \sum \vec{p}_{\mathtt{v}^*,\mathrm{bot}}. \tag{A9}$$

The total number of faults is $n_0 + n_{\mathrm{on}} + n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* + n_{\mathtt{cap}} \leq d-1$, which means that $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d - 1 - n_0 - n_{\mathrm{on}} - n_{\mathtt{cap}}$ (where $n_{\mathtt{cap}} \geq 1$). Consider the following subcases.

(3.a) If $n_0 \geq 1$ or $n_{\mathrm{on}} \geq 1$ or $n_{\mathtt{cap}} \geq 2$, then $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d-3$. In this case, Eqs. (A7)–(A9) contradict Condition 2 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault, and $\mathtt{v}$ and $\mathtt{v}^*$ faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(3.b) If $n_0 = 0, n_{\mathrm{on}} = 0$, and $n_{\mathtt{cap}} = 1$, we find that $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* + n_{\mathtt{cap}} \leq d-1$ and $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d-2$. Let us divide the proof into the following further subcases (where some subcases may overlap).

(i) If $n_{\mathtt{v}} + n_{\mathtt{v}}^* = 0$, then Eqs. (A8) and (A9) contradict Condition 0 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault).

(ii) If $n_{\mathtt{v}} + n_{\mathtt{v}}^* = 1$, then Eqs. (A7)–(A9) contradict Condition 3 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault, and $\mathtt{v}$ and $\mathtt{v}^*$ faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(iii) If $n_{\mathtt{off}} = 0$, then $n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d-2$. In this case, Eqs. (A7)–(A9) contradict Condition 1 (where $\mathtt{v}$ and $\mathtt{v}^*$ faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(iv) If $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* + n_{\mathtt{cap}} \leq d-2$ (or, equivalently, $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d-3$), then Eqs. (A7)-(A9) contradict Condition 2 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault, and $\mathtt{v}$ and $\mathtt{v}^*$ faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

(v) If $n_{\mathtt{off}} \geq 1$, $n_{\mathtt{v}} + n_{\mathtt{v}}^* \geq 2$, and $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* + n_{\mathtt{cap}} = d-1$, then Eqs. (A1), (A2), (A5), and (A7)–(A9) contradict Condition 5 (where $\mathtt{q}_{\mathtt{off}}$, $\mathtt{v}$, $\mathtt{v}^*$, $\mathtt{cap}$ faults correspond to $\mathtt{q}_{\mathtt{2D}}$, $\mathtt{f}_{\mathtt{2D}}$, $\mathtt{v}^*_{\mathtt{2D}}$, and $\mathtt{cap}_{\mathtt{2D}}$ faults, respectively).

Case 4: $n_{\mathtt{f}} \geq 1$ and $n_{\mathtt{cap}} \geq 1$. (The main equations cannot be simplified in this case.) From the fact that the total number of faults is at most $d-1$, we have $n_{\mathtt{off}} + n_{\mathtt{v}} + n_{\mathtt{v}}^* \leq d-3$. In this case, we find that Eqs. (A7)–(A9) contradict Condition 2 (where a $\mathtt{q}_{\mathtt{off}}$ fault corresponds to a $\mathtt{q}_{\mathtt{2D}}$ fault, and $\mathtt{v}$ and $\mathtt{v}^*$ faults correspond to an $\mathtt{f}_{\mathtt{2D}}$ fault).

So far, we have shown that if Conditions 1–5 are satisfied and all faults give rise to purely Z-type errors, then there is no fault combination arising from up to $d-1$ faults whose combined error is a logical Z operator and its cumulative flag vector is zero. Because the circuits for each pair of X-type and Z-type generators use the same CNOT ordering, the same analysis is also applicable to the case of purely X-type errors; i.e., if Conditions 1–5 are satisfied and all faults give rise to purely X-type errors, then there is no fault combination arising from up to $d-1$ faults whose combined error is a logical X operator and its cumulative flag vector is zero. In the next part of the proof, we use these results to show that $\mathcal{F}_t$ is distinguishable.

Let us consider a fault combination whose combined error is of mixed type. Let t_x and t_z denote the total numbers of faults during the measurements of X-type and Z-type generators, and let u_x, u_y, u_z denote the numbers of qubit faults that give X-type, Y-type, and Z-type errors, respectively. Suppose that the fault combination arises from no more than $d-1$ faults; we have $t_x + t_z + u_x + u_y + u_z \leq d-1$. Next, observe that t_x faults during the measurement of X-type generators cannot cause a Z-type error of weight more than t_x, and that t_z faults during the measurement of Z-type generators cannot cause an X-type error of weight more than t_z. Thus, the Z part of the combined error and the cumulative flag vector corresponding to Z-type generators can be considered as an error and a cumulative flag vector arising from $t_z + t_x + u_z + u_y \leq$

$d-1$ faults that give rise to purely Z-type errors. Similarly, the X part of the combined error and the cumulative flag vector corresponding to X-type generators can be considered as an error and a cumulative flag vector arising from $t_x+t_z+u_x+u_y \leq d-1$ faults that give rise to purely X-type errors. Recall that there is no fault combination arising from up to $d-1$ faults whose combined error is a logical X (or a logical Z) operator and its cumulative flag vector is zero when all faults give rise to purely X-type (or purely Z-type) errors. Using this, we find that, for any fault combination arising from $d-1$ faults, it cannot correspond to a nontrivial logical operator and the zero cumulative flag vector. That is, there is no fault combination corresponding to a nontrivial logical operator and the zero cumulative flag vector in $\mathcal{F}_{2t}$, where $2t=d-1$. By Proposition 1, this implies that $\mathcal{F}_t$ is distinguishable.

APPENDIX B: FAULT-TOLERANT ERROR-CORRECTION PROTOCOL FOR A GENERAL STABILIZER CODE

In Sec. V B, we constructed a FTEC protocol for a capped color code in H form of any distance in which its fault set is distinguishable. We also showed that such a protocol is fault tolerant when the r filter, the ideal decoder, and the distinguishable error set are defined as in Definitions 12, 13, and 15. Using similar ideas, we can also construct a FTEC protocol for a general stabilizer code whose circuits for the syndrome measurement give a distinguishable fault set $\mathcal{F}_t$, i.e., a code in which $\mathcal{E}_r$ is defined by Definition 11 instead of Definition 15. The outcome bundle defined for the protocol in this section is similar to the outcome bundle defined for the FTEC protocol for a capped color code, except that the syndrome $\vec{\mathbf{s}}$ and the cumulative flag vector $\vec{\mathbf{f}}$ are not separated into X and Z parts. We can also build a list of all possible fault combinations and their corresponding combined error and cumulative vector from the distinguishable fault set $\mathcal{F}_t$. The FTEC protocol for a general stabilizer code is as follows.

FTEC protocol for a stabilizer code whose syndrome measurement circuits give a distinguishable fault set. During a single round of full syndrome measurement, measure the all generators in any order. Perform full syndrome measurements until the outcome bundles $(\vec{\mathbf{s}},\vec{\mathbf{f}})$ are repeated $t+1$ times in a row. Afterwards, do the following.

1. Determine an EC operator F using the list of possible fault combinations as follows.
   (a) If there is a fault combination on the list whose syndrome and cumulative flag vector are $\vec{\mathbf{s}}$ and $\vec{\mathbf{f}}$, then F is the combined error of such a fault combination. (If there is more than one fault combination corresponding to $\vec{\mathbf{s}}$ and $\vec{\mathbf{f}}$, a combined error of any of such fault combinations will work since they are logically equivalent.)
   (b) If none of the fault combinations on the list corresponds to $\vec{\mathbf{s}}$ and $\vec{\mathbf{f}}$, then F can be any Pauli operator whose syndrome is $\vec{\mathbf{s}}$.
2. Apply F to the data qubits to perform error correction.

To verify that the FTEC protocol for a general stabilizer code satisfies both properties of a FTEC gadget according to the revised definition (Definition 14), we can use an analysis similar to that presented in Sec. V B, except that $\mathcal{E}_r$ is defined by Definition 11 instead of Definition 15 and the errors in the analysis (E_{in}, E_a, and E_b) need not be separated into X and Z parts.

[1] T. Tansuwannont and D. Leung, Fault-tolerant quantum error correction using error weight parities, Phys. Rev. A **104**, 042410 (2021).
[2] P. W. Shor, Fault-tolerant quantum computation, *Proc., 37th Annu. Symp. Found. Comput. Sci.*, p. 56, (1996).
[3] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, SIAM J. Comput., (2008).
[4] A. Y. Kitaev, Quantum computations: Algorithms and error correction, Russ. Math. Surv. **52**, 1191 (1997).
[5] E. Knill, R. Laflamme, and W. H. Zurek, Threshold accuracy for quantum computation, (1996), ArXiv:quant-ph/9610011.
[6] J. Preskill, Reliable quantum computers, Proc. R. Soc. London Ser. A: Math., Phys. Eng. Sci. **454**, 385 (1998).
[7] B. M. Terhal and G. Burkard, Fault-tolerant quantum computation for local non-Markovian noise, Phys. Rev. A **71**, 012336 (2005).
[8] M. A. Nielsen and C. M. Dawson, Fault-tolerant quantum computation with cluster states, Phys. Rev. A **71**, 042323 (2005).
[9] P. Aliferis and D. W. Leung, Simple proof of fault tolerance in the graph-state model, Phys. Rev. A **73**, 032308 (2006).
[10] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2006).
[11] A. M. Steane, Overhead and noise threshold of fault-tolerant quantum error correction, Phys. Rev. A **68**, 042322 (2003).
[12] A. Paetznick and B. W. Reichardt, Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit Golay code, Quantum Inf. Comput. **12**, 1034 (2012).
[13] C. Chamberland, T. Jochym-O'Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, Phys. Rev. A **95**, 022313 (2017).
[14] R. Takagi, T. J. Yoder, and I. L. Chuang, Error rates and resource overheads of encoded three-qubit gates, Phys. Rev. A **96**, 042302 (2017).

[15] D. P. DiVincenzo and P. Aliferis, Effective Fault-Tolerant Quantum Computation with Slow Measurements, Phys. Rev. Lett. **98**, 020501 (2007).
[16] E. Knill, Scalable quantum computing in the presence of large detected-error rates, Phys. Rev. A **71**, 042322 (2005).
[17] A. M. Steane, Active Stabilization, Quantum Computation, and Quantum State Synthesis, Phys. Rev. Lett. **78**, 2252 (1997).
[18] A. M. Steane, Fast fault-tolerant filtering of quantum codewords, (2002), arXiv preprint ArXiv:quant-ph/0202036.
[19] T. J. Yoder and I. H. Kim, The surface code with a twist, Quantum **1**, 2 (2017).
[20] R. Chao and B. W. Reichardt, Quantum Error Correction with Only Two Extra Qubits, Phys. Rev. Lett. **121**, 050502 (2018).
[21] R. Chao and B. W. Reichardt, Flag Fault-Tolerant Error Correction for any Stabilizer Code, PRX Quantum **1**, 010302 (2020).
[22] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).
[23] T. Tansuwannont, C. Chamberland, and D. Leung, Flag fault-tolerant error correction, measurement, and quantum computation for cyclic Calderbank-Shor-Steane codes, Phys. Rev. A **101**, 012342 (2020).
[24] 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).
[25] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and subsystem codes on low-degree graphs with flag qubits, Phys. Rev. X **10**, 011022 (2020).
[26] R. Chao and B. W. Reichardt, Fault-tolerant quantum computation with few qubits, npj Quantum Inf. **4**, 42 (2018).
[27] C. Chamberland and A. W. Cross, Fault-tolerant magic state preparation with flag qubits, Quantum **3**, 143 (2019).
[28] Y. Shi, C. Chamberland, and A. Cross, Fault-tolerant preparation of approximate GKP states, New J. Phys. **21**, 093007 (2019).
[29] P. Baireuther, M. Caio, B. Criger, C. W. Beenakker, and T. E. O'Brien, Neural network decoder for topological color codes with circuit level noise, New J. Phys. **21**, 013003 (2019).
[30] A. Bermudez, X. Xu, M. Gutiérrez, S. Benjamin, and M. Müller, Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources, Phys. Rev. A **100**, 062307 (2019).
[31] C. Vuillot, Is error detection helpful on IBM 5q chips?, Quantum Inf. Comput. **18**, 0949 (2018).
[32] M. Gutiérrez, M. Müller, and A. Bermúdez, Transversality and lattice surgery: Exploring realistic routes toward coupled logical qubits with trapped-ion quantum processors, Phys. Rev. A **99**, 022330 (2019).
[33] L. Lao and C. G. Almudever, Fault-tolerant quantum error correction on near-term quantum processors using flag and bridge qubits, Phys. Rev. A **101**, 032333 (2020).
[34] D. M. Debroy and K. R. Brown, Extended flag gadgets for low-overhead circuit verification, Phys. Rev. A **102**, 052409 (2020).
[35] A. Rodriguez-Blanco, A. Bermudez, M. Müller, and F. Shahandeh, Efficient and Robust Certification of Genuine Multipartite Entanglement in Noisy Quantum Error Correction Circuits, PRX Quantum **2**, 020304 (2021).
[36] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).
[37] G. Duclos-Cianci and D. Poulin, Fast Decoders for Topological Quantum Codes, Phys. Rev. Lett. **104**, 050504 (2010).
[38] S. Bravyi and J. Haah, Quantum Self-Correction in the 3D Cubic Code Model, Phys. Rev. Lett. **111**, 200501 (2013).
[39] G. Duclos-Cianci and D. Poulin, Kitaev's Z d-code threshold estimates, Phys. Rev. A **87**, 062338 (2013).
[40] H. Anwar, B. J. Brown, E. T. Campbell, and D. E. Browne, Fast decoders for qudit topological codes, New J. Phys. **16**, 063038 (2014).
[41] S. Bravyi, M. Suchara, and A. Vargo, Efficient algorithms for maximum likelihood decoding in the surface code, Phys. Rev. A **90**, 032326 (2014).
[42] N. Delfosse, Decoding color codes by projection onto surface codes, Phys. Rev. A **89**, 012317 (2014).
[43] B. J. Brown, D. Loss, J. K. Pachos, C. N. Self, and J. R. Wootton, Quantum memories at finite temperature, Rev. Mod. Phys. **88**, 045005 (2016).
[44] B. J. Brown, N. H. Nickerson, and D. E. Browne, Fault-tolerant error correction with the gauge color code, Nat. Commun. **7**, 1 (2016).
[45] C. Chamberland and P. Ronagh, Deep neural decoders for near term fault-tolerant experiments, Quantum Sci. Technol. **3**, 044002 (2018).
[46] K. Duivenvoorden, N. P. Breuckmann, and B. M. Terhal, Renormalization group decoder for a four-dimensional toric code, IEEE Trans. Inf. Theory **65**, 2545 (2018).
[47] A. S. Darmawan and D. Poulin, Linear-time general decoding algorithm for the surface code, Phys. Rev. E **97**, 051302 (2018).
[48] A. Kubica and N. Delfosse, Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders, (2019), arXiv preprint ArXiv:1905.07393.
[49] A. Kubica and J. Preskill, Cellular-Automaton Decoders with Provable Thresholds for Topological Codes, Phys. Rev. Lett. **123**, 020501 (2019).
[50] N. Maskara, A. Kubica, and T. Jochym-O'Connor, Advantages of versatile neural-network decoding for topological codes, Phys. Rev. A **99**, 052351 (2019).
[51] N. H. Nickerson and B. J. Brown, Analysing correlated noise on the surface code using adaptive decoding algorithms, Quantum **3**, 131 (2019).
[52] M. Vasmer, D. E. Browne, and A. Kubica, Cellular automaton decoders for topological quantum codes with noisy measurements and beyond, Sci. Rep. **11**, 1 (2021).
[53] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary, (1998), arXiv preprint ArXiv:quant-ph/9811052.
[54] H. Bombin and M. A. Martin-Delgado, Topological Quantum Distillation, Phys. Rev. Lett. **97**, 180501 (2006).
[55] 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).

[56] T. Karzig, C. Knapp, R. M. Lutchyn, P. Bonderson, M. B. Hastings, C. Nayak, J. Alicea, K. Flensberg, S. Plugge, and Y. Oreg, *et al.*, Scalable designs for quasiparticle-poisoning-protected topological quantum computation with Majorana zero modes, Phys. Rev. B **95**, 235305 (2017).

[57] R. Chao, M. E. Beverland, N. Delfosse, and J. Haah, Optimization of the surface code design for Majorana-based qubits, Quantum **4**, 352 (2020).

[58] A. Kubica, B. Yoshida, and F. Pastawski, Unfolding the color code, New J. Phys. **17**, 083026 (2015).

[59] M. Vasmer and D. E. Browne, Three-dimensional surface codes: Transversal gates and fault-tolerant architectures, Phys. Rev. A **100**, 012312 (2019).

[60] B. Eastin and E. Knill, Restrictions on Transversal Encoded Quantum Gate Sets, Phys. Rev. Lett. **102**, 110502 (2009).

[61] S. Bravyi and R. König, Classification of Topologically Protected Gates for Local Stabilizer Codes, Phys. Rev. Lett. **110**, 170503 (2013).

[62] F. Pastawski and B. Yoshida, Fault-tolerant logical gates in quantum error-correcting codes, Phys. Rev. A **91**, 012305 (2015).

[63] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. Sloane, Quantum Error Correction and Orthogonal Geometry, Phys. Rev. Lett. **78**, 405 (1997).

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

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

[66] A. Steane, Multiple-particle interference and quantum error correction, Proc. R. Soc. London Ser. A: Math., Phys. Eng. Sci. **452**, 2551 (1996).

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

[68] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).

[69] G. Nebe, E. M. Rains, and N. J. Sloane, The invariants of the Clifford groups, Des., Codes Cryptogr. **24**, 99 (2001).

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

[71] A. Paetznick and B. W. Reichardt, Universal Fault-Tolerant Quantum Computation with Only Transversal Gates and Error Correction, Phys. Rev. Lett. **111**, 090505 (2013).

[72] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, Fault-Tolerant Conversion between the Steane and Reed-Muller Quantum Codes, Phys. Rev. Lett. **113**, 080501 (2014).

[73] H. Bombín, Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes, New J. Phys. **17**, 083002 (2015).

[74] A. Kubica and M. E. Beverland, Universal transversal gates with color codes: A simplified approach, Phys. Rev. A **91**, 032330 (2015).

[75] 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).

[76] D. Poulin, Stabilizer Formalism for Operator Quantum Error Correction, Phys. Rev. Lett. **95**, 230504 (2005).

[77] D. Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, Phys. Rev. A **73**, 012340 (2006).

[78] 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).

[79] S. Bravyi and A. Cross, Doubled color codes, (2015), arXiv preprint ArXiv:1509.03239.

[80] C. Jones, P. Brooks, and J. Harrington, Gauge color codes in two dimensions, Phys. Rev. A **93**, 052332 (2016).

[81] D. Gottesman, Class of quantum error-correcting codes saturating the quantum Hamming bound, Phys. Rev. A **54**, 1862 (1996).

[82] A bare logical operator is a logical operator that acts on the logical qubit(s) of a subsystem code and does not affect the gauge qubit(s); see Refs. [76,77,85].

[83] A perfect code is a quantum code that saturates the quantum Hamming bound; i.e., there is a one-to-one correspondence between correctable errors and all possible syndromes [67, 81]. A perfect CSS code is defined similarly, except that the syndromes of X-type and Z-type errors are considered separately.

[84] H. Bombín, Single-shot fault-tolerant quantum error correction, Phys. Rev. X **5**, 031043 (2015).

[85] S. Bravyi, Subsystem codes with spatially local generators, Phys. Rev. A **83**, 012320 (2011).