Fault-Tolerant Code-Switching Protocols for Near-Term Quantum Processors

Friederike Butt,[1,2,*] Sascha Heußen,[1,2] Manuel Rispler,[1,2] and Markus Müller[1,2]

[1] *Institute for Quantum Information, RWTH Aachen University, Aachen, Germany*

[2] *Institute for Theoretical Nanoelectronics (PGI-2), Forschungszentrum Jülich, Jülich, Germany*

(Received 18 August 2023; revised 14 February 2024; accepted 10 April 2024; published 28 May 2024)

Topological color codes are widely acknowledged as promising candidates for fault-tolerant quantum computing. Neither a two-dimensional nor a three-dimensional topology, however, can provide a universal gate set {H, T, CNOT}, with the T gate missing in the two-dimensional and the H gate in the three-dimensional case. These complementary shortcomings of the isolated topologies may be overcome in a combined approach, by switching between a two- and a three-dimensional code while maintaining the logical state. In this work, we construct resource-optimized deterministic and nondeterministic code-switching protocols for two- and three-dimensional distance-three color codes using fault-tolerant quantum circuits based on flag qubits. Deterministic protocols allow for the fault-tolerant implementation of logical gates on an encoded quantum state, while nondeterministic protocols may be used for the fault-tolerant preparation of magic states. Taking the error rates of state-of-the-art trapped-ion quantum processors as a reference, we find a logical failure probability of 3% for deterministic logical gates, which cannot be realized transversally in the respective code. By replacing the three-dimensional distance-three color code in the protocol for magic state preparation with the morphed code introduced in Vasmer and Kubica [PRX Quantum 3, 030319 (2022)], we reduce the logical failure rates by 2 orders of magnitude, thus rendering it a viable method for magic state preparation on near-term quantum processors. Our results demonstrate that code switching enables the fault-tolerant and deterministic implementation of a universal gate set under realistic conditions, and thereby provide a practical avenue to advance universal, fault-tolerant quantum computing and enable quantum algorithms on first, error-corrected logical qubits.

DOI: 10.1103/PRXQuantum.5.020345

I. INTRODUCTION

Universal quantum computation holds the promise to perform certain computational tasks exponentially faster than any known classical algorithm [1,2]. In the current noisy intermediate-scale quantum (NISQ) era, however, the accuracy of quantum algorithms is limited by noise [3]. A prospective means of increasing their robustness against the noise is to encode quantum information on logical qubits, with each logical qubit consisting of multiple physical qubits. On these logical qubits, quantum error correction (QEC) [4,5] can be performed to correct for errors on physical qubits and, thereby, to recover the initially encoded information [6,7]. For physical error rates below a certain threshold, QEC enables practical quantum computing for arbitrarily long times given suitable, i.e., fault-tolerant (FT), circuit constructions [8–11]. FT circuits can be designed by using *transversal* gate operations, which are composed of single-qubit unitaries acting on individual qubits in each encoded block, such that potential errors in any of these operations are prevented from proliferating uncontrollably [5,12]. In order to approximate arbitrary computations on encoded qubits, a discrete set of operations that forms a *universal* gate set can be used, which requires at least one non-Clifford gate [8,13]. However, the Eastin-Knill theorem states that no QEC code exists that has a universal, transversally encoded and, therefore, FT gate set [14]. This complicates the implementation of a universal FT gate set and poses a key challenge towards error-corrected universal quantum computing.

Recent quantum computing experiments are focused on the practical encoding of qubits on a logical level and investigate the implementation of a quantum memory. For example, on trapped-ion systems, the preparation of logical states [15], error-detecting codes [16], FT stabilizer readouts [17] in a shuttling-based architecture [18], as well as repeated cycles of QEC [19] have been implemented.

*Corresponding author: friederike.butt@rwth-aachen.de

In superconducting qubits, logical qubits have successfully been initialized [20,21], while repeated QEC has been realized [22–24], as well as error-detection codes [25,26]. Recently, a distance-five surface-code logical qubit outperformed a distance-three logical qubit, demonstrating an improvement of performance of QEC codes with an increasing number of qubits [27]. Rydberg atoms are a promising candidate for building quantum processors due to their strong long-range interactions and scalability [28,29] and have shown rapid progress in single- and multiqubit control [30–32] as well as the first elements of quantum error correction [33]. Advancements in other qubit architectures have been reported, as, for example, a three-qubit phase-correcting code in a silicon-based architecture [34] and fault-tolerant operations on a logical qubit using spin qubits in diamond [35], among others.

These advancements in practical and scalable implementations of logical qubits enable FT operations on these encoded states and motivate the search for ways to achieve FT, universal quantum computing on near-term devices. The FT control of an error-corrected single logical qubit [36] has been demonstrated on a trapped-ion processor, as well as logical operations in a distance-two error-detecting surface code on a superconducting architecture [37] and entangling gates between logical qubits [38]. A universal set of gates was recently implemented for the first time on a seven-qubit Steane code, which is the smallest error-correcting color code, using FT circuit constructions with flag qubits [39]. Here, the universal gate set is completed by using magic state injection to realize a logical non-Clifford operation. The non-Clifford T gate can be implemented by preparing a magic state fault tolerantly on an auxiliary system [40,41] and then injecting this magic state onto the target qubits using a logical CNOT gate [42].

An alternative to magic state injection is to complete a FT universal gate set using code switching between codes with complementary transversal gate sets [43–45]. Code switching allows one to transfer encoded information between specific codes. To do so, stabilizers are measured to project the state onto the desired codespace. Based on the measurement outcomes, local Pauli operations are applied to change the state into the correct $+1$ eigenstate of the stabilizers of the target code. One candidate for implementing a universal gate set with code switching are two-dimensional color codes, because all Clifford gates can be implemented transversally [46–49]. For a universal set of gates, at least one non-Clifford gate such as the T gate is required, which can only be implemented transversally in three-dimensional codes, as, for example, tetrahedral color codes as illustrated in Fig. 1 [50]. By switching between these two- and three-dimensional color codes in a FT manner, it is possible to access all gates of a universal gate set with a transversal implementation.

To achieve fault tolerance for code-switching protocols, it is not sufficient to simply use FT stabilizer measurements for error correction. The objective for syndrome measurements for error correction is to detect errors on data qubits and correct for them. The objective of stabilizer measurements during code switching is to project the state onto a desired codespace and apply a corresponding switching operation. In this case, single errors on data qubits invert the measurement outcomes and can directly introduce logical errors on the target code. Furthermore, the initial code can have different error-correcting properties than the target code, which has to be carefully taken into account, as we discuss in detail in Sec. V.

In this work, we present strategies that allow for FT switching between two- and three-dimensional color codes. We explicitly consider the distance-three instances of these codes. These instances consist of a number of physical qubits that is available on near-term quantum processors while a single arbitrary computational error is correctable. We make use of auxiliary flag qubits that herald the presence of errors resulting from faults in the circuits [40,41,51]. The concept of flag qubits has been extended to codes with arbitrary distance [47,52,53] and has been used for the FT initialization of logical qubits [39–41] and the FT encoding of magic states [54]. We develop FT protocols

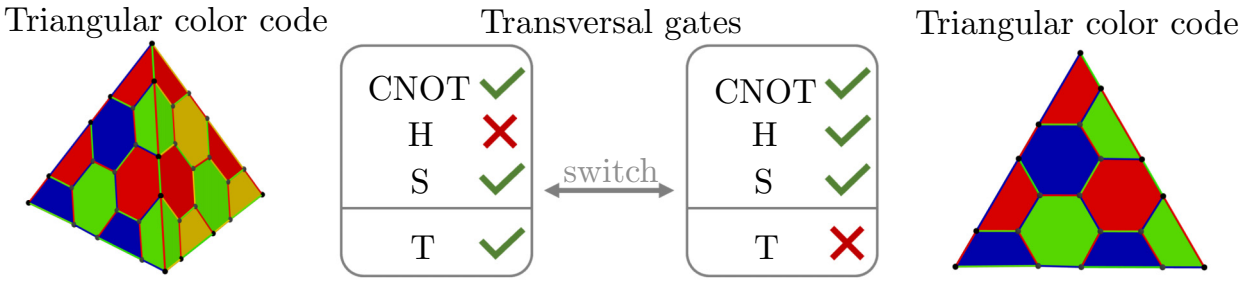

FIG. 1. Transversal gate set of three-dimensional tetrahedral and two-dimensional triangular color codes. A two-dimensional triangular color code, as depicted on the right, has a transversal implementation of the CNOT and the Hadamard gate. On the left side, a three-dimensional tetrahedral code is illustrated, which has a transversal implementation of the CNOT gate and the non-Clifford T gate. By switching between these two code classes, it is possible to access all gates of a universal gate set {CNOT, H, T} with transversal implementations.

considering circuit-level noise, i.e., noise models where all components of the underlying circuits, such as initialization, gate operations, and measurements are modeled as noisy. We construct new flag-qubit-based circuits for the encoding of logical states and for sequences of stabilizer measurements, which are resource optimized in terms of qubit and gate count. Furthermore, we introduce new nondeterministic code-switching protocols that make use of a transformed *morphed* code, which was introduced in Ref. [55]. The morphed three-dimensional distance-three color code inherits the FT T gate and makes it a candidate for completing a universal gate set. We construct a scheme for the FT preparation of a magic state using this new code. In doing so, the success rate can be increased significantly compared to the preparation of a magic state via the initial three-dimensional color code. Using the morphed code for magic state preparation, we find similar success rates as state-of-the-art implementations [39]. We construct an entire toolbox of modular FT building blocks, which are illustrated in Fig. 2. Each block is FT by itself and different blocks can be composed to FT protocols. Table I summarizes the resources for the constructed building blocks and composite protocols, illustrated in Fig. 2, in terms of the two-qubit gate count and the number of qubits.

This paper is structured as follows. In Secs. II and III we briefly review basic properties of two- and three-dimensional color codes and, specifically, the smallest instances of these code classes. In Sec. IV, we summarize the underlying theory for code switching. In Sec. V, we discuss the different Pauli errors that have to be considered in detail and present strategies for a FT implementation of code switching. We introduce FT switching with the morphed code in Sec. VI and present results of numerical simulations of noisy circuits implementing logical

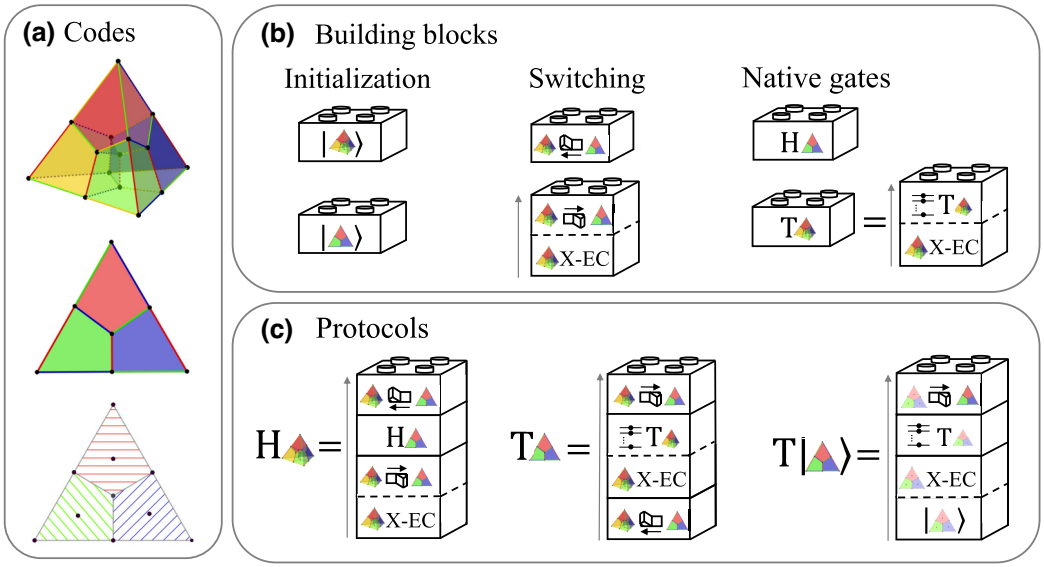

FIG. 2. Codes and FT building blocks used to compose FT code-switching protocols. (a) The tetrahedral $[\![15,1,3]\!]$ code (top) contains 15 physical qubit, encodes one logical qubit and has distance three. The two-dimensional $[\![7,1,3]\!]$ Steane code (center) consists of seven physical qubits and has the same distance and number of logical qubits. The morphed $[\![10,1,2]\!]$ code (bottom) is made up of ten physical qubits, encodes a single logical qubit, but is a distance-two code, so arbitrary single errors are detectable, yet not correctable. When constructing code-switching schemes, we distinguish between deterministic and nondeterministic protocols. Deterministic protocols are implemented using the tetrahedral code, correcting for any single error in each block. Nondeterministic protocols include postselecting during switching and replacing the tetrahedral code with the error-detecting morphed $[\![10,1,2]\!]$ code. (b) FT building blocks for the considered codes. These include the unitary initialization of logical states on the considered codes (left). The unitary encoding circuits are given in Figs. 20–23. The central blocks represent FT switching to and from the $[\![7,1,3]\!]$ Steane code. The X-EC block represents one round of X-error correction, which is required when switching to the two-dimensional code. In the illustrated blocks, each tetrahedron can be replaced by the morphed $[\![10,1,2]\!]$ code to obtain the corresponding block for the morphed code. (c) The FT building blocks illustrated in (b) can be combined into FT protocols. The logical FT Hadamard gate for the tetrahedral $[\![15,1,3]\!]$ code (left) can be implemented by, first, switching to the Steane code, then, applying the transversal Hadamard gate and, lastly, switching back to the tetrahedral code. Similarly, the T gate for the $[\![7,1,3]\!]$ Steane code (center) can be realized by switching to the three-dimensional code, applying the transversal T gate and switching back to the Steane code afterwards. These two protocols correspond to deterministic FT logical operations, which are not restricted to specific input states. The right column represents the preparation of a magic state on the $[\![7,1,3]\!]$ Steane code with the morphed $[\![10,1,2]\!]$ code. After initializing the logical $|\overline{+}\rangle$ state on the morphed code, a reduced round of X-error detection is applied. Then, a FT T gate is applied to the morphed code, followed by FT switching to the $[\![7,1,3]\!]$ Steane code.

TABLE I. Required resources for FT building blocks used for composing FT code-switching protocols as illustrated in Fig. 2. The center column indicates the number of qubits, which are required for a given block. The right column contains the number of CNOT gates that have to be implemented for the specified protocol. It limits the circuit depth and is an indication for the experimental circuit complexity. The number of CNOT gates is specified for the case that no error occurs during the given protocol, which in the limit of low physical error rates corresponds to the most likely case. If errors occur, this number can increase or decrease the number of required two-qubit gates. For example, if a flag qubit heralds the presence of an error, a different set of stabilizers has to be measured afterwards, which changes the number of CNOT gates. The number of CNOT gates required for X-error correction on the 15-qubit tetrahedral code is much higher than for any other block, since all of the ten Z stabilizers of the tetrahedral code have to be measured twice with flags to achieve fault tolerance.

	No. qubits	No. CNOT gates
(a) $[\![15,1,3]\!] \rightarrow [\![7,1,3]\!]$	17	18
(a) X-EC for $[\![15,1,3]\!]$	17	120
(a) $[\![10,1,2]\!] \rightarrow [\![7,1,3]\!]$	12	18
(a) X-EC for $[\![10,1,2]\!]$	12	42
(a) $[\![7,1,3]\!] \rightarrow [\![15,1,3]\!]$	17	72
(a) $[\![7,1,3]\!] \rightarrow [\![10,1,2]\!]$	12	34
(b) $\lvert\bar{0}\rangle$ for $[\![10,1,2]\!]$	11	15
(b) $\lvert\overline{+}\rangle$ for $[\![10,1,2]\!]$	10	14
(b) $\lvert\bar{0}\rangle$ for $[\![15,1,3]\!]$	16	25
(b) $\lvert\overline{+}\rangle$ for $[\![15,1,3]\!]$	16	32
(b) initialization bulk	10	20
(c) T gate on $[\![15,1,3]\!]$	15	0
(c) T gate on $[\![10,1,2]\!]$	10	12
(c) MS with $[\![10,1,2]\!]$	12	40

operations in Sec. VIII for a single-parameter noise model. In Sec. IX, we consider a multiparameter noise model and estimate the projected performance on near-term devices, specifically focusing on state-of-the-art trapped-ion quantum processors. Lastly, we provide conclusions and an outlook in Sec. X.

II. 2D COLOR CODES

Two-dimensional topological color codes were originally proposed by Bombin *et al.* [46]. They fall into the important class of so-called CSS (Calderbank-Shor-Steane) stabilizer codes [56,57]. They can be specified by the code parameters $[\![n,k,d]\!]$ where n is the number of physical qubits, k is the number of encoded logical qubits, and d is the code distance. The simultaneous $+1$ eigenspace of all stabilizer generators corresponds to the codespace spanned by the valid logical states [43]. A two-dimensional color code can be constructed by placing qubits on the vertices of a three-valent and three-colorable lattice [46]. Different vertices are connected by *links*. *Faces* are formed by a closed set of links. Three colors, say red, blue, and green $\{R,B,G\}$ are commonly assigned to the faces. Colors are assigned to links according to the color of the faces they connect, so that, for example, blue faces are connected by blue links. Colors are assigned in such a way that three links of different color meet at each vertex. By closing the boundaries periodically to form a torus, one can identify the logical operators of the two-dimensional color codes as the noncontractible loops on the torus.

A closely related class of two-dimensional color codes with open boundary conditions are triangular color codes, as exemplarily illustrated in Fig. 1. They can be constructed by placing the described lattice on the surface of a sphere and removing one vertex and all its neighboring links and faces. This lattice contains an odd number of physical qubits and can be deformed into a triangular shape. X and Z stabilizers are defined on the faces of the code, which ensures that all stabilizers overlap at an even number of vertices and, therefore, commute. Logical operators can be implemented by applying X and Z operations to all qubits. By applying stabilizers, one finds equivalent string-type logical operators with a minimum length d, which correspond to the length of one edge of the triangle. Triangular codes encode one logical qubit and have a transversal implementation of {H, CNOT, S}. This means that the logical $\overline{\text{CNOT}}$ can be realized by applying physical CNOT gates to pairs of qubits on two codes. The Hadamard gate $\overline{\text{H}}$ can be implemented by applying a single H gate to each physical qubit. It interchanges HX H $= Z$ on each individual qubit and, therefore, acts as a Hadamard gate on the logical level since the logical X and Z operators have the same support. It analogously maps X stabilizers to Z stabilizers and vice versa and, therefore, $\overline{\text{H}}$ preserves the stabilizer group. Similarly, the phase gate S can be realized transversally by applying single qubit S and $\text{S}^\dagger$ [43]. With the set of gates {H, S, CNOT}, the whole Clifford group of gates can be generated transversally [58]. The smallest triangular color code with this set of transversal gates is a code that is equivalent to the Steane code [59].

A. The Steane code $[\![7,1,3]\!]$

The $[\![7,1,3]\!]$ Steane code consists of $n=7$ physical qubit, encodes $k=1$ logical qubit and has distance $d=3$ [59]. Six stabilizer generators are defined as

$$\begin{aligned} S_R^X &= X_0X_1X_2X_3, \quad S_R^Z = Z_0Z_1Z_2Z_3 \\ S_G^X &= X_1X_2X_4X_5, \quad S_G^Z = Z_1Z_2Z_4Z_5 \\ S_B^X &= X_2X_3X_5X_6, \quad S_B^Z = Z_2Z_3Z_5Z_6 \end{aligned} \tag{1}$$

with the color labels red (R), green (G), and blue (B) and for the indexing given in Fig. 3(a). The two logical operators can be implemented by applying Pauli X and Z operators to all seven qubit and are stabilizer equivalent to operators of minimum weight 3. For example, the logical

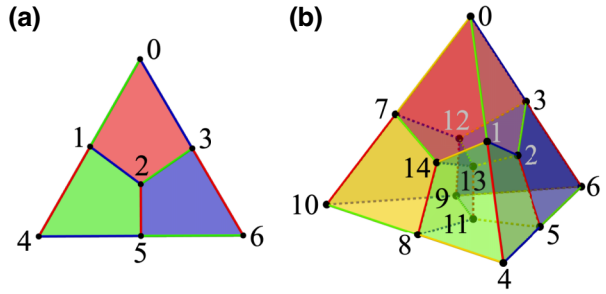

FIG. 3. Illustration of the two- and three-dimensional distance-three color codes. (a) The $[\![7,1,3]\!]$ Steane code consists of seven qubits (black dots) forming a green, red, and blue face (colored faces). The X- and Z-type stabilizers S^{σ}_{col} of color $\{R,G,B\}$ are defined as $\sigma = X, Z$ Pauli operators applied to the qubits that form the given face. (b) The 15-qubit tetrahedral code $[\![15,1,3]\!]$ has distance three, encodes one logical qubit and consists of 15 qubits (black dots), which form three-dimensional units. Four X stabilizers B^X_c are defined on the cells $c \in \{R,B,G,Y\}$ of the code, as, for example, the red X stabilizer $B^X_R = X_0X_1X_2X_3X_7X_{12}X_{13}X_{14}$. Ten independent Z stabilizers B^Z_f are defined on the faces $f \in \{R,B,G,Y,RB,RG,RY,BG,BY,GY\}$ of the code, as, for example, the red-yellow Z stabilizer $B^Z_{RY} = Z_7Z_{12}Z_{13}Z_{14}$.

Pauli operators can be implemented on the edges of the triangle with

$$\overline{X} = X_4X_5X_6 \quad \text{and} \quad \overline{Z} = Z_4Z_5Z_6. \tag{2}$$

III. 3D COLOR CODES

To construct a three-dimensional color code, qubits are arranged on a three-dimensional, four-valent, and four-colorable lattice structure. Four colors red, green, blue, and yellow $\{R,G,B,Y\}$ are assigned to the smaller three-dimensional units, which are called *cells*. Cells of the same color never touch but are connected by links of that same color, for example, blue cells are connected by blue links [50]. So, a link has color $\kappa \in \{R,B,G,Y\}$ if the three cells, of which the link is part of, have colors different from κ. The two-dimensional boundaries of the cells are called *faces* and have the color labels $\kappa_1\kappa_2$ according to the colors of the two cells of which they are part of Ref. [60]. This construction can be generalized also to n-dimensional codes [43,44].

One class of three-dimensional color codes are tetrahedral codes, as exemplarily illustrated in Fig. 1. Analogously to the two-dimensional case, a tetrahedral code can be constructed by arranging a three-dimensional lattice within the volume of a 3 sphere, in the above-described manner. By removing one vertex and its neighboring cells, faces and links, the lattice contains an odd number of physical qubits and can be deformed into a tetrahedron, encoding a single logical qubit [50]. One associates faces F with Z stabilizers and cells C with X stabilizers, ensuring commutativity since each cell and face contains an even number of qubits by construction. A codestate $|\overline{\psi}\rangle$ of this system is characterized by the conditions

$$S^X_c\,|\overline{\psi}\rangle = |\overline{\psi}\rangle \quad \forall c \in C, \tag{3}$$

$$S^Z_f\,|\overline{\psi}\rangle = |\overline{\psi}\rangle \quad \forall f \in F. \tag{4}$$

An operator basis is defined for the qubit encoded in the tetrahedral code by

$$\overline{X} = X^{\otimes n} \quad \text{and} \quad \overline{Z} = Z^{\otimes n} \tag{5}$$

with the total number of qubits n. This definition ensures that all stabilizers commute with the logical Pauli operators, since the tetrahedral code contains an odd number of physical qubits and each stabilizer has support on an even number of vertices. The $\overline{\mathrm{T}}$ gate

$$\overline{\mathrm{T}} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}. \tag{6}$$

can be implemented transversally on specific tetrahedral codes if the following two conditions are fulfilled. Let M be subset of the set of all physical qubits Q. Then let T gates be applied to the subset M and $\mathrm{T}^\dagger$ gates applied to the subset $M^C = Q \setminus M$. Then for the support of the X stabilizer G, the transversal $\overline{\mathrm{T}}$ gate can be implemented in this manner if [43,60,61]

(1) $|M \cap G| - |M^C \cap G| = 0 \bmod 8$.
In words, an equal number of T and $\mathrm{T}^\dagger$ gates has to be applied in each cell (mod8).
(2) $|M| - |M^C| = 1 \bmod 8$.
In words, T has to be applied to one more qubit than $\mathrm{T}^\dagger$ in total (mod8).

It has been shown that the tetrahedral $[\![15,1,3]\!]$ code is the smallest distance-three QEC code with a transversal non-Clifford gate [62].

A. The 15-qubit tetrahedral code $[\![15,1,3]\!]$

The 15-qubit tetrahedral code $[\![15,1,3]\!]$, as illustrated in Fig. 3(b), consists of 15 physical qubits, one encoded logical qubit and can correct for one arbitrary error [43]. Four X-type stabilizers B^X_c are defined on the cells

$$c \in \{R,B,G,Y\} \tag{7}$$

of the code, as depicted in Fig. 3(b). Six independent Z-type stabilizers B^Z_f are defined on the faces within the tetrahedron $f \in \{RB,RG,RY,BG,BY,GY\}$, and four on the boundary of the tetrahedron $f \in \{R,B,G,Y\}$. Any choice

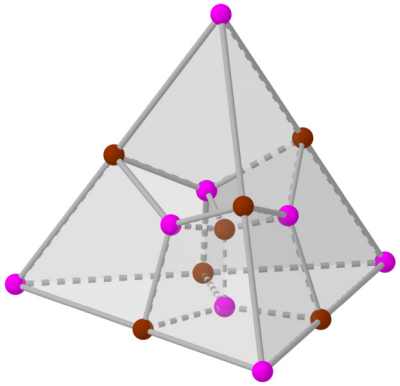

FIG. 4. Bipartition for the implementation of a transversal T gate on the tetrahedral $[\![15,1,3]\!]$ code. The vertices in the $[\![15,1,3]\!]$ code can be partitioned into two sets, so that vertices connected by an edge belong to different sets [60]. Given this kind of bipartition, a $\overline{\mathrm{T}}$ gate can be implemented transversally by applying single T gates to the qubits on the pink positions and $\mathrm{T}^\dagger$ to qubits on the brown positions.

of the three faces on the boundary of each cell generates the same stabilizer group. The logical operators are defined as

$$\overline{X} = X^{\otimes 15} \quad \text{and} \quad \overline{Z} = Z^{\otimes 15}. \tag{8}$$

These are stabilizer equivalent to logical X operators with minimum weight 7 and logical Z operators with minimum weight 3. Accordingly, the tetrahedral $[\![15,1,3]\!]$ code has distance $d_x = 7$ for X errors and distance $d_z = 3$ for Z errors, so it is possible to correct X errors of weight less or equal to 3 and any single Z error, as well as error configurations that are stabilizer equivalent to these. The $\overline{\mathrm{CNOT}}$ gate can be realized in a transversal manner by coupling two code blocks pairwise with physical CNOT gates. The H gate cannot be implemented transversally (note that the stabilizer group is in particular not preserved under the operation $\mathrm{H}^{\otimes n}$). In Fig. 4, we show a bipartition that implements a transversal T gate according to the above stated conditions.

IV. CODE SWITCHING

Encoded information can be transferred between specific stabilizer codes, if they correspond to two variants of the same subsystem code (subsystem codes also known as gauge codes). A subsystem code is defined by its gauge group $\mathcal{G}$, describing a general subgroup of the n-qubit Pauli group on a set of physical qubits [63,64]. The stabilizer group $\mathcal{S} \subseteq \mathcal{G}$ is the center of $\mathcal{G}$, which is generated by those Pauli operators in $\mathcal{G}$ that commute with all elements in $\mathcal{G}$. A stabilizer code can be viewed as a special case of a subsystem code where $\mathcal{S} = \mathcal{G}$. The subspace of codestates can be split into a tensor product of the logical qubits and so-called gauge qubits [43]

$$|\psi\rangle = |\overline{\psi}\rangle \otimes |g\rangle_G . \tag{9}$$

Here, $|\overline{\psi}\rangle$ represents the logical state and $|g\rangle_G$ represents the gauge state. The gauge state corresponds to extra degrees of freedom, which are not uniquely fixed by the stabilizers. An element U of $\mathcal{G} \setminus \mathcal{S}$ has the effect on this state $|\psi\rangle$

$$U|\psi\rangle = |\overline{\psi}\rangle \otimes U_G |g\rangle_G , \tag{10}$$

leaving the logical state $|\overline{\psi}\rangle$ unchanged and affecting only the gauge state $|g\rangle_G$. Thereby, equivalent codestates can be generated by applying elements of $\mathcal{G} \setminus \mathcal{S}$ without changing the encoded information. A logical gate $\overline{L}$ can affect both the logical state and the gauge state, while preserving the codespace [43]

$$\overline{L}|\psi\rangle = \left(\overline{L'}|\overline{\psi}\rangle\right) \otimes |g'\rangle_G . \tag{11}$$

Given the stabilizer group $\mathcal{S}_A$ of code A, for which $\mathcal{S} \subset \mathcal{S}_A$, any $+1$ eigenstate of $\mathcal{S}_A$ is consequentially also a codestate in the subsystem. Therefore, any codestate $|\psi\rangle_A \in \mathcal{S}_A$ is also a subsystem codestate, i.e., it can be written as a tensor product of a logical state and a gauge state as in Eq. (9)

$$|\psi\rangle_A = |\overline{\psi}\rangle \otimes |g_A\rangle_G . \tag{12}$$

Analogously, a second code B defined by stabilizer group $\mathcal{S}_B$ can be defined, for which $\mathcal{S} \subset \mathcal{S}_B$, so we can write

$$|\psi\rangle_B = |\overline{\psi'}\rangle \otimes |g_B\rangle_G . \tag{13}$$

The stabilizer groups of the three codes considered are illustrated in Fig. 5. If the logical operators $\overline{L}$, $\overline{L}_A$, and $\overline{L}_B$ of the three codes can be represented in the same way, and $|\psi\rangle_A$ and $|\psi\rangle_B$ correspond to the same logical state in their stabilizer code, they must be logically equivalent in the

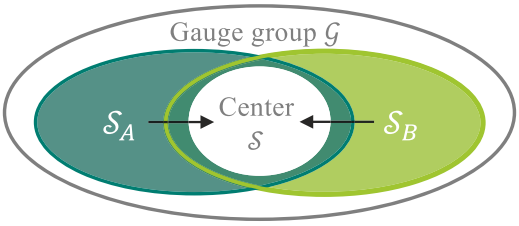

FIG. 5. Illustration of the stabilizer groups of two stabilizer codes A and B within the same subsystem code. The stabilizer groups of a stabilizer code $\mathcal{S}_A$ and $\mathcal{S}_B$ are part of the gauge group of the corresponding subsystem code $\mathcal{G}$ with its center $\mathcal{S}$. Both codes $\mathcal{S}_A$ and $\mathcal{S}_B$ contain $\mathcal{S}$. Therefore, codestates in A and B can be written as tensor products in the subsystem $|\psi\rangle_A = |\overline{\psi}\rangle \otimes |g_A\rangle_G$ and $|\psi\rangle_B = |\overline{\psi'}\rangle \otimes |g_B\rangle_G$.

subsystem code. Mathematically, this means that $|\overline{\psi}\rangle$ and $|\overline{\psi'}\rangle$ are the same logical state for both

$$|\psi\rangle_A = |\overline{\psi}\rangle \otimes |g_A\rangle_G \quad \text{and} \quad |\psi\rangle_B = |\overline{\psi'}\rangle \otimes |g_B\rangle_G . \tag{14}$$

So, in the subsystem code, $|\psi\rangle_A$ and $|\psi\rangle_B$ differ only in their gauge state $|g_A\rangle$, $|g_B\rangle$. Elements of $\mathcal{G} \setminus \mathcal{S}$ affect only the gauge state and leave the logical state unchanged. These elements can be used to *fix* the gauge state from $|g_A\rangle$ to $|g_B\rangle$ or vice versa, while leaving the logical state unchanged.

Now, let us consider switching from code A to code B. First, those stabilizers of the target code B, which are not fulfilled initially, have to be measured. This measurement projects the state randomly into a ± 1 eigenstate of the measured stabilizer. Secondly, elements of $\mathcal{G} \setminus \mathcal{S}$ are applied, which only affect the gauge state, to force the gauge state into the corresponding state $|\psi\rangle_B = |\overline{\psi}\rangle \otimes |g_B\rangle_G$.

A. Code switching between the $[\![7,1,3]\!]$ and $[\![15,1,3]\!]$

Switching between the Steane code and the 15-qubit tetrahedral code has been discussed and analyzed in Refs. [44,61,65], among others. In the following, we briefly review non-FT switching between these codes before presenting a FT scheme in Sec. V.

One can define a subsystem code whose gauge group $\mathcal{G}$ is generated by all independent X and Z faces that can be defined on the tetrahedral structure. The stabilizer group of this subsystem code $\mathcal{S}_{\text{subsystem}}$ is generated by those elements that commute with all other elements of $\mathcal{G}$ and is therefore generated by the X and Z cells of the tetrahedron. In contrast to the stabilizers of the stabilizer code $[\![15,1,3]\!]$, the Z stabilizers of the subsystem code are not defined on the ten independent faces of the code, but only on the four weight-8 cells. The stabilizer group of the subsystem is part of both the stabilizer group of the Steane code, together with the bulk, and of the stabilizer group of the $[\![15,1,3]\!]$ tetrahedral code, meaning

$$\begin{aligned} \mathcal{S}_{\text{subsystem}} &\subset (\mathcal{S}_{\text{Steane}+\text{bulk}}), \\ \mathcal{S}_{\text{subsystem}} &\subset \mathcal{S}_{\text{tetrahedron}}. \end{aligned} \tag{15}$$

Here, the bulk is formed by those data qubits of the tetrahedron that are not part of the Steane code (yellow cell). Note that, if the state fulfills two opposing faces individually, then it is also a $+1$ eigenstate of its products, which corresponds to the composed cell. Consequentially, every codestate of the tetrahedral stabilizer code, which is a $+1$ eigenstate of all elements of $\mathcal{S}_{\text{tetra}}$, is always also a codestate of the subsystem code. Analogously, every codestate of the Steane code together with the bulk is also always a codestate of the subsystem. So, one can write codestates of the $[\![15,1,3]\!]$ code and the $[\![7,1,3]\!]$ Steane code in the subsystem as a tensor product of the corresponding logical state and a specific gauge state, as given in Eq. (14).

If the logical operators of the three codes have a common representation, so that $|\overline{\psi}\rangle$ is the same logical state for both codes, they must be logically equivalent in the subsystem code. One common representation for the logical operators of the Steane code and the $[\![15,1,3]\!]$ tetrahedral code corresponds to one side of the tetrahedron, such as

$$\overline{X} = X_0X_1X_2X_3X_4X_5X_6 \quad \text{and} \quad \overline{Z} = Z_0Z_1Z_2Z_3Z_4Z_5Z_6 \tag{16}$$

for the indexing given in Fig. 3(b). To switch between the two codes $[\![7,1,3]\!]$ and $[\![15,1,3]\!]$, first, stabilizers of the target code have to be measured. This projects the encoded state into a $+1$ or -1 eigenstate of the measured stabilizers. For a negative measurement outcome, a combination of gauge operators has to be applied that forces the state into a $+1$ eigenstate of all stabilizers of the target code.

Concretely, to switch from the $[\![15,1,3]\!]$ code to the Steane code, first, one has to measure the three X faces of the Steane code, which are $S_R^X = X_0X_1X_2X_3$, $S_B^X = X_2X_3X_5X_6$ and $S_G^X = X_1X_2X_4X_5$ for the indexing given in Fig. 3(b). Based on these measurement outcomes, a combination of Z faces connecting the Steane code with the bulk is applied in order to force the state into the codespace of the Steane code. For example, let us consider the case where the measurement outcome of the X faces of the Steane code is $(S_R^X, S_B^X, S_G^X) = (1,0,0)$. This means that the measurement projected the state onto a $+1$ eigenstate of the blue and green faces and a -1 eigenstate of the red face. In order to force the state into the desired codespace, we apply the gauge operator $B_{BG}^Z = Z_2Z_5Z_{11}Z_{13}$, as illustrated in Fig. 7(a). This operator overlaps at an even number of vertices with the blue and green face, so the state stays a $+1$ eigenstate of these stabilizers. It overlaps at a single vertex with the red face and, therefore, forces the state into a $+1$ eigenstate of this stabilizer. Note that this is in contrast to using single Pauli operations as one would do for correcting errors on a set of qubits. This would affect both the gauge state $|g\rangle_G$ and the logical state $|\overline{\psi}\rangle$, as given in Eq. (9) and, therefore, change the encoded information.

This protocol is inverted for the other direction. To switch from the Steane code to the $[\![15,1,3]\!]$ code, first, one has to measure the three Z faces that connect the Steane code and the bulk, which are $B_{RB}^Z = Z_2Z_3Z_{12}Z_{13}$, $B_{RG}^Z = Z_1Z_2Z_{13}Z_{14}$, and $B_{BG}^Z = Z_2Z_5Z_{11}Z_{13}$. Based on these measurement outcomes, a combination of the Steane X faces is applied in order to change the state into the codespace of the $[\![15,1,3]\!]$ code.

V. DETERMINISTIC FT CODE SWITCHING

For the physical implementation of code-switching protocols on quantum processors, the underlying operations

have to be constructed in a way that is resilient against noise. In this section, we aim at finding schemes that tolerate any single error on an arbitrary component of the circuit. Then, the probability that two components of the circuit fail at the same time and thus cause a logical failure is order p^2, where p is the failure probability for a single component [5]. For deterministic schemes, we should be able to correct any error arising from a single faulty component in the circuit, which includes faulty measurements, initializations, single- and two-qubit operations. In order to construct these FT code-switching protocols, a set of criteria has to be fulfilled.

The first criterion is that the error configuration on the initial code also has to be correctable on the target code. For the color codes considered, it is possible to correct weight-3 X errors on the three-dimensional code, while it is only possible to correct a single X error on the two-dimensional code. Starting in a codestate of the three-dimensional code that contains a weight-3 error configuration, this would initially be correctable. Assuming perfect switching, which does not introduce any additional errors, this error configuration would be transferred onto the two-dimensional code, where a weight-3 error configuration can directly correspond to a logical operator. Therefore, we perform one round of X-error correction with flag qubits before switching to the two-dimensional code. This corrects any previously present configuration with up to three X errors and corrects for any single error that occurs during the EC procedure.

The second condition is that we have to be able to identify errors on data qubits. Specific errors on data qubits invert the outcome of the stabilizer measurements. If we cannot distinguish this inverted measurement outcome from the original information, we do not notice that an error has occurred and a logical error can be introduced, as illustrated in Fig. 7. Therefore, one does not directly achieve fault tolerance for code switching by simply using the concept of FT stabilizer measurements with flag qubits. Considering realistic circuitry with noisy components, it is necessary to consider these errors on data qubits as well.

In general, we can obtain the expectation value of an operator $\overline{O}$, that has a transversal implementation on a given code, by coupling an auxiliary qubit to the data qubits that participate in the measurement [66]. Using this kind of circuit to extract the syndrome is, however, not FT because single faults can directly cause a logical failure. A single fault on

(A) auxiliary qubits during a stabilizer measurement,
(B) data qubits before or during a stabilizer measurement,
(C) a measurement of an auxiliary qubit at the end of a stabilizer readout

can induce a logical error, which we discuss in detail in the remainder of this section.

A. Errors on auxiliary qubits

Single errors on the auxiliary qubit can propagate onto data qubits. An X error after the second CNOT gate in the circuit shown in Fig. 6(a) propagates onto two data qubits. On the target Steane code, this results in a logical failure, since in the Steane code, only a single X error is correctable. Analogously, Z errors on the auxiliary qubit during the measurement of a Z stabilizer can lead to a logical error $\overline{Z}$ on the target code, since only a single Z error is correctable in the $[\![15, 1, 3]\!]$ tetrahedral stabilizer code. We can implement stabilizer measurements, which are FT with

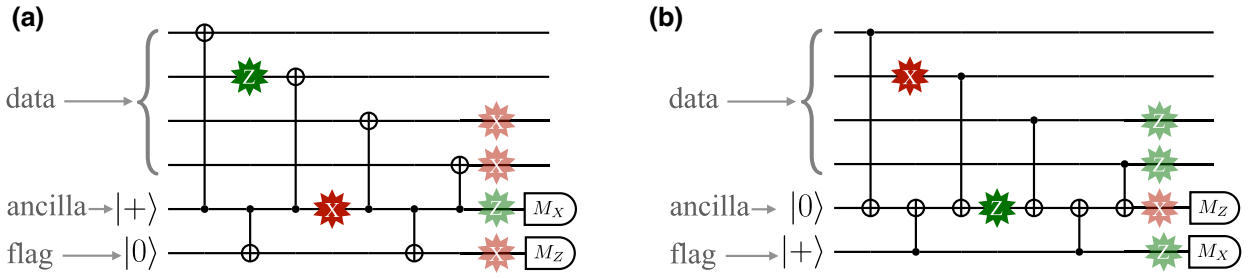

FIG. 6. Circuits for stabilizer measurements using flag qubits [51]. (a) Circuit for measuring a weight-4 Pauli-X operator containing an auxiliary qubit, which is coupled to the data qubits. An additional flag qubit is coupled with CNOT gates to the first auxiliary qubit. An X error in the middle of the circuit, as depicted in red, propagates onto two data qubits and the flag qubit. Measuring the flag qubit in the end indicates that a potentially dangerous error has occurred. By performing an additional Z-face stabilizer measurement without flags afterwards, one can detect these errors and correct for them. Since an error before the second or after the next to last CNOT gate results only in an error, which is equivalent to a weight-1 error on the data qubits, this scheme is FT with respect to errors on the auxiliary qubits. A Z error during this X-stabilizer measurement, as depicted in green, propagates onto the auxiliary qubit, which changes the measured syndrome as illustrated in Fig. 7. In order to avoid applying a logical error, these errors on the data qubits have to be detected and corrected. (b) Circuit for measuring a weight-4 Pauli-Z operator using an additional flag qubit. Analogously, Z errors on the auxiliary qubits are detected by the flag qubit and can be corrected by performing an additional stabilizer measurement. X errors on the data qubits during the Z-face measurement can propagate and cause a logical failure.

respect to these errors on auxiliary qubits, using a *flag*-based measurement scheme, as depicted in Fig. 6. The concept of flag qubits was introduced in Ref. [51] and has, for example, been used for FT error correction in color codes in two and three dimensions [47,52,53] and the initialization of logical qubits [39–41]. Again, an auxiliary qubit is coupled to the data qubits via CNOT gates and a flag qubit is coupled to the auxiliary qubit. If a dangerous fault occurs on the auxiliary qubit during this measurement, it still propagates onto one or more of the data qubits, but it also propagates onto the flag qubit. In that case, the flag qubit will be measured in the -1 state in the end and indicates that a potentially dangerous error has occurred.

If a circuit flags, meaning that a flag qubit was measured in the corresponding -1 eigenstate, we still have to identify which error has occurred. The circuit could have flagged owing to a measurement error in the end, or to an error before the second to last CNOT gate. Specifically, we have to check if the flag error leads to a logical failure. Here, these dangerous flag errors correspond to propagated weight-2 errors on the data qubits resulting from a single error on the auxiliary qubit. In order to localize these errors, one additional Z (X) stabilizer has to be measured. After this additional measurement of a single stabilizer, the complete syndrome is measured again without flag qubits. Together with the information of which circuit flagged before, the dangerous weight-2 errors can be identified and corrected.

B. Errors on data qubits

Errors on the data qubits can propagate onto the auxiliary qubit and invert the measurement outcome, as illustrated in Fig. 6. Taking this inverted measurement outcome would cause us to apply the incorrect switching operation and can lead to a logical failure on the target code.

For example, a single Z error on the top qubit 0 would invert the measurement outcome of the red face, when switching from the tetrahedral code to the Steane code. We would then apply the according switching operation, as illustrated in Fig. 7(a). But owing to the inverted measurement outcome, we now introduce additional errors on the data qubits of the Steane code that overlap with the applied gauge operator. Together with the initial error, this can directly correspond to a logical operator on the Steane code. For FT code switching, these errors on data qubits have to be corrected.

To identify and correct for these errors on data qubits, one can exploit that the state should always be a $+1$ eigenstate of the complete cells. This means that, for switching from the $[\![15,1,3]\!]$ code to the $[\![7,1,3]\!]$ Steane code, the X faces opposite of the measured Steane faces have to agree with the measured Steane X face to also fulfill the cells. For example, considering the red cell, as illustrated in Fig. 3, the state is either in a $+1$ eigenstate of both $S_R^X = X_0X_1X_2X_3$ and $B_{RY}^X = X_7X_{12}X_{13}X_{14}$ or in a -1 eigenstate of both these operators so that the state is a $+1$ eigenstate of the corresponding cell, which is formed by these two opposing faces [61]. If all three pairs of opposing faces agree, we know that either no data error or an error on the corner qubit of the yellow cell (qubit 10) has occurred and we can proceed. If we find some disagreement, we know that an error has happened on a specific pair of qubits. To localize the error, we measure the yellow X cell. If we measure -1, we know that an the error has occurred on the corresponding qubit of the yellow cell and can correct for it and continue. If we find $+1$, we know that the error has occurred on the Steane code and update the syndrome accordingly. In practice, the bulk, which is

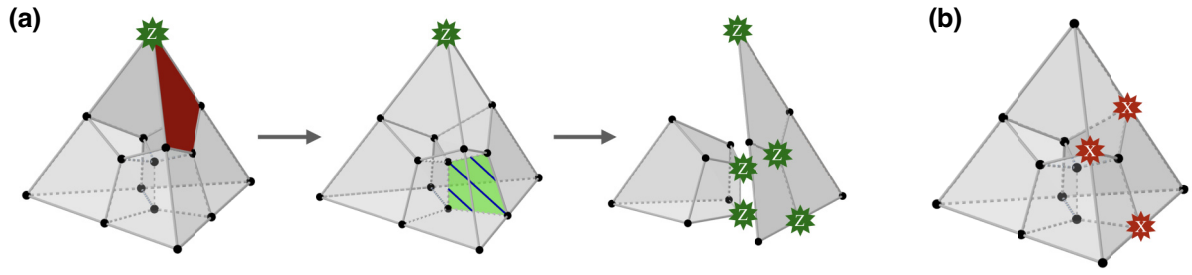

FIG. 7. Error configurations on data qubits that can lead to a logical failure. (a) Consider switching from the 15-qubit tetrahedral code to the target Steane code, which lives on the seven qubits forming the right three-face triangle of the tetrahedron. To transfer the encoded information to the Steane code, we measure the three X faces of the Steane code. If, for example, a Z error has occurred on the top qubit of the red cell (large green position on the left) before the measurement, this inverts the measurement outcome of the corresponding face. If we originally had measured the syndrome $(S_R^X, S_B^X, S_G^X) = (0,0,0)$, we would now find $(S_R^X, S_B^X, S_G^X) = (1,0,0)$. Based on this measurement outcome, we would apply B_{BG}^Z to switch to the Steane code (center). If we then consider the Steane code (right), we effectively applied Pauli-Z operations to the three qubits on the large green positions. This directly corresponds to a logical $\overline{Z}$ on the Steane code and results in a logical failure. (b) The three positions where a single X error leads to a logical failure when switching from the Steane code to the tetrahedral code are depicted in red.

formed by those data qubits of the tetrahedron that are not part of the Steane code (yellow cell), can be measured destructively to localize all potentially dangerous errors on the data qubits.

For switching back to the tetrahedral code afterwards, the bulk has to be in a specific state that corresponds to the $+1$ eigenstate of its cells and faces. To this end, the bulk has to be reinitialized into the correct state using the circuit given in Fig. 22. For switching from the Steane code to the $[\![15, 1, 3]\!]$ code, we identify only three qubit positions on which a single X error can lead to a logical failure indicated in Fig. 7(b), by numerically checking all possible error positions. We can distinguish these X errors by measuring the complete Z syndrome of the Steane code. To avoid further error propagation, we use the flag qubit scheme for this syndrome measurement. If a circuit flags, we measure one of the weight-8 X cells using a single auxiliary qubit to localize dangerous flag errors. After identifying the dangerous flag error, we remeasure the syndrome once without flags, to ensure that the syndrome contains the detected error. We can then update the syndrome according to the detected error correctly and continue.

C. Measurement errors

A single error on the measurement of the first auxiliary qubit in the circuits in Fig. 6 would invert the measurement outcome and can lead to a logical failure after applying the corresponding switching operation. To achieve fault tolerance, we repeat the syndrome measurement and take a majority vote. If we obtain the same syndrome twice, we assume that no single measurement error has occurred and continue. If the syndromes are different, we repeat the syndrome measurement a third time, without a flag qubit, and proceed with the result of this last measurement. The probability that the measurement is incorrect 2 times is order p^2, where p is the failure probability for a single measurement [5]. Note that for switching from $[\![15, 1, 3]\!]$ to $[\![7, 1, 3]\!]$ it is sufficient to measure only the target stabilizers once, because any measurement error is identified by checking the agreement of opposing faces as explained in the previous subsection. The complete FT protocols are summarized in Appendix A.

VI. NONDETERMINISTIC FT CODE SWITCHING

The above deterministic schemes for code switching require a large number of gates, as summarized in Table I, and a large circuit depth due the additional checks that have to be performed. We propose nondeterministic schemes as more feasible alternatives on near-term devices. These schemes achieve lower failure rates than the deterministic protocols, due to effectively postselecting for errors and reducing the required number of CNOT gates and, therefore, the circuit depth. However, postselecting results in a finite success rate, since a fraction of runs is discarded. These nondeterministic schemes allow, for example, the preparation of a magic state on a two-dimensional color code with fewer resources than for the deterministic scheme while achieving lower logical failure rates. The methods we describe in the following are postselection and code morphing to an effective error-detecting code.

A. Code switching with postselection

For FT switching between the Steane and the $[\![15, 1, 3]\!]$ code, we check for potentially dangerous errors on data and auxiliary qubits. As discussed in Sec. V, this is done by measuring extra stabilizers in addition to the gauge operators and using flag qubits. We now adjust the scheme to postselect for any detected error: whenever an error is detected, we now stop the protocol and discard the results. This includes flag qubits measured in the $|1\rangle$ or $|-\rangle$ state and any disagreement in opposing faces for the data error check. By postselecting [67,68] for any detected error, a fraction of weight-2 or higher weight errors is sorted out, resulting in lower logical failure rates. The number of required qubits and two-qubit gates remains the same as for the fault-free deterministic protocols, since the exact same circuits are executed.

B. Code switching with morphed codes

Code morphing was introduced in Ref. [55] and is a method for generating new codes from existing ones by effectively turning the logical qubits of a subcode, called the *child* code, into bare physical qubits. This new code inherits the logical gates of the *parent* code. By morphing a suitable code, it is possible to find a code that still has a fault-tolerant T gate, but requires fewer qubits and has lower stabilizer weights, while the code distance is reduced. Therefore, the number of required CNOT gates for the additional stabilizer measurements for error checks can be reduced. This offers a more feasible alternative for code switching on current experimental setups.

C. Morphing the tetrahedral code $[\![15, 1, 3]\!]$

The $[\![15, 1, 3]\!]$ code contains a smaller stabilizer code on, for example, the yellow cell of the code. This smaller subcode corresponds to a $[\![8, 3, 2]\!]$ code and is called the child code. This $[\![8, 3, 2]\!]$ code has a transversal implementation of the CCZ gate [69].

One can invert the encoding circuit of the $[\![8, 3, 2]\!]$ code, by reverting the order of gates and the direction of all CNOT gates, and apply it to the tetrahedral code. Thereby, the three encoded qubits of the $[\![8, 3, 2]\!]$ are replaced by bare, physical qubits, leaving the new morphed $[\![10, 1, 2]\!]$ code, as illustrated in Fig. 8 and discussed in Appendix B. The new stabilizers of the morphed code can be derived by replacing the logical operators of the child code with the corresponding operators acting on the bare physical qubit.

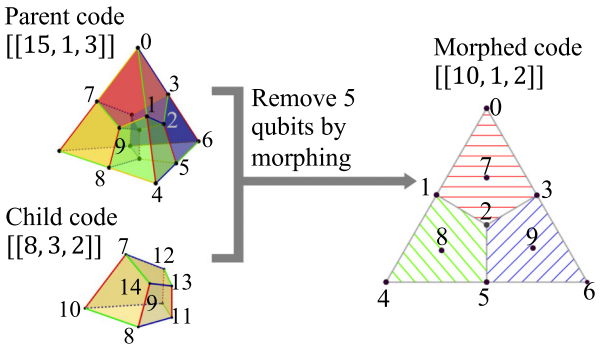

FIG. 8. Morphing the tetrahedral $[\![15,1,3]\!]$ into the $[\![10,1,2]\!]$ error-detecting code. One can morph the tetrahedral $[\![15,1,3]\!]$ code with the yellow cell $[\![8,3,2]\!]$, contained in the initial parent code. By taking the encoding circuit of the $[\![8,3,2]\!]$ child code and applying its inverse to a codestate of the $[\![15,1,3]\!]$ parent code, five qubits are decoupled while keeping the encoded information in the remaining qubits. Morphing effectively reduces the number of qubits and the stabilizer weights while reducing the distance. Three physical qubits of the yellow cell remain in the final morphed code, which are depicted in the center of the faces as qubits 7, 8, and 9. We obtain the $[\![10,1,2]\!]$ code, which has new stabilizers of lower weight than the tetrahedral parent code.

The previous weight-8 X stabilizers of the parent code are turned into weight-5 cells on the morphed $[\![10,1,2]\!]$ code. We illustrate the weight-5 operators by placing one extra qubit in the center of the faces of the Steane code, as depicted in Fig. 8 on the right. These are

$$\begin{aligned} B_R'^X &= X_0X_1X_2X_3X_7 \\ B_B'^X &= X_2X_3X_5X_6X_9 \\ B_G'^X &= X_1X_2X_4X_5X_8. \end{aligned} \tag{17}$$

Three Z stabilizers are defined on the faces of the code, and three Z stabilizers are defined on the extended face intersections as

$$\begin{aligned} B_R'^Z &= Z_0Z_1Z_2Z_3, \quad B_{GB}'^Z = Z_2Z_5Z_7 \\ B_B'^Z &= Z_2Z_3Z_5Z_6, \quad B_{RG}'^Z = Z_1Z_2Z_9 \\ B_G'^Z &= Z_1Z_2Z_4Z_5, \quad B_{RB}'^Z = Z_2Z_3Z_8. \end{aligned} \tag{18}$$

The lookup table for the morphed $[\![10,1,2]\!]$ code is given in the Appendix. By morphing a code, we can effectively trade-off different code properties. In this case, we traded-off the reduced stabilizer weights against code distance, since the morphed code is only an error-detecting code with distance $d'=2$. Note that code morphing is not physically executed but is a strategy to generate a new code that can be then used for code-switching protocols.

The logical $\overline{\mathrm{T}}$ gate can be implemented on the $[\![10,1,2]\!]$ code using the circuit shown in Fig. 9(a). This implementation is not transversal anymore but fault tolerant in the sense that all possible errors resulting from a single Z error can be detected. All possible X errors on qubits 7, 8, and 9, as, for example, $X_7X_8X_9$, are correctable because the corresponding Z syndromes do not coincide with any single-error configuration. Therefore, we can assign the syndromes to these higher weight X errors and correct for them. This implementation of $\overline{\mathrm{T}}$ can be understood by interpreting code morphing as a replacement of the logical qubits, encoded in the yellow cell, by three bare physical qubits. The action of the T and $\mathrm{T}^\dagger$ gates in the FT implementation of the logical $\overline{\mathrm{T}}$ gate on the yellow cell corresponds to a logical $\overline{\mathrm{CCZ}}$ on the three logical qubits of the child code $[\![8,3,2]\!]$ [69]. By morphing the tetrahedral code, this logical operation $\overline{\mathrm{CCZ}}$ is replaced by the corresponding action on the three bare physical qubits, which is the CCZ gate on 7, 8, and 9. The CCZ gate itself can be implemented using only single T, $\mathrm{T}^\dagger$, and CNOT gates as shown in Fig. 9(b). The morphed $[\![10,1,2]\!]$ code is the smallest known stabilizer code that has a FT T gate.

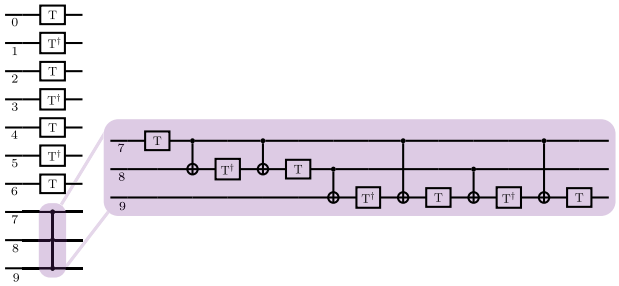

FIG. 9. Implementation of the FT logical $\overline{\mathrm{T}}$ gate for the $[\![10,1,2]\!]$ code [55]. Single T and $\mathrm{T}^\dagger$ gates are applied to qubits 0–6 and a CCZ operation is applied to the three qubits in the center of the faces. The CCZ gate on qubits 7, 8, and 9 can be decomposed into single-qubit and CNOT gates [70] using the implementation given on the right. This implementation of $\overline{\mathrm{T}}$ with a three-qubit gate is still fault tolerant even though errors on qubits 7, 8, or 9 can propagate onto other data qubits. Any Z error on these qubits is detectable and all possible X errors on these qubits are correctable.

D. FT code switching between $[\![10,1,2]\!]$ and $[\![7,1,3]\!]$

In order to switch between the two stabilizer codes, first, the two codes must have a common representation of the logical operators and, secondly, a subsystem code with stabilizer group $\mathcal{S}$ has to be defined that suffices $\mathcal{S} \subset \mathcal{S}_{[\![\infty\prime,\infty,\in]\!]}$ and $\mathcal{S} \subset (\mathcal{S}_{\mathrm{Steane}}, \mathcal{S}_{\mathrm{bulk}})$. The $[\![10,1,2]\!]$ code inherits the logical operators from its parent code, the 15-qubit tetrahedral code. Since the 15-qubit tetrahedral code and the Steane code have a common representation of their logical operators, this condition is consequentially also fulfilled for the $[\![10,1,2]\!]$ code. Furthermore, a suitable subsystem can be defined whose gauge group $\mathcal{G}$ is generated by the stabilizers of the $[\![10,1,2]\!]$ code, which

includes the Z faces of the Steane code, and the single-qubit Pauli operators X_7, X_8, and X_9 for the indexing given in Fig. 8. The stabilizer group $\mathcal{S}$ of the subsystem is then generated by the weight-5 X-and weight-4 Z faces of the $[\![10,1,2]\!]$ code structure.

We can switch from the $[\![10,1,2]\!]$ code to the Steane code by measuring the stabilizers of the Steane code, which are not fulfilled by the initial codestate in the $[\![10,1,2]\!]$ code. These stabilizers are the X faces of the Steane code, since the $[\![10,1,2]\!]$ code only fulfills the weight-5 X cells and not the weight-4 X faces. After measuring the stabilizers of the target code, we apply combinations of the gauge operators $B'^{Z}_{BG} = Z_2Z_5Z_7$, $B'^{Z}_{RG} = Z_1Z_2Z_9$, and $B'^{Z}_{RB} = Z_2Z_3Z_8$, in order to fix the gauge state accordingly.

The FT-switching scheme from the 15-qubit tetrahedral code to the Steane code, discussed in the previous section, can directly be adapted for the morphed code. The X faces of the Steane code are measured using flag qubits and errors on data qubits can, again, be detected by measuring opposing pairs of stabilizers in the subsystem. We need only to replace the bulk measurement with single-qubit measurements of qubits 7, 8, and 9 in the X basis. If some disagreement between opposing pairs of stabilizers, as, for example, between X_7 and $B'^{X}_{R} = X_0X_1X_2X_3$, is found, we have to discard the corresponding run because Z errors on the data qubits cannot be uniquely identified in the subsystem, which makes the protocol nondeterministic.

For switching from the Steane code to the $[\![10,1,2]\!]$ code, we have to measure those stabilizers of the $[\![10,1,2]\!]$ code, which are not fulfilled in the Steane code. These are the weight-3 Z stabilizers on the extended intersections of faces $(B'^{Z}_{RB}, B'^{Z}_{GB}, B'^{Z}_{RG})$. Based on the measurement outcome, we apply combinations of the Steane X faces. Since the stabilizers are of weight 3, any error on the auxiliary qubit during these measurements may indeed propagate but is only equivalent to a single error on the data qubits. So, for a FT measurement of the weight-3 stabilizers, we do not need additional flag qubits. To check for errors on the data qubits that can lead to a logical failure, we measure the green and blue Z faces. If some error is discovered in this check, we discard this run. If no error is found, we take the measured Z syndrome and continue. These measurements can also be performed without additional flag qubits, since every dangerous flag error is detectable on the target code.

In the following section, we present simulation results for code-switching protocols using the tetrahedral $[\![15,1,3]\!]$ and the morphed $[\![10,1,2]\!]$ code.

VII. ERROR MODEL AND SIMULATION METHODS

In the following simulations, we investigate the logical failure rates p_L for different protocols by performing Monte Carlo (MC) simulations. We consider circuit-level noise with a single-error parameter p, which describes the error rates on each component. This includes single- and two-qubit gates, as well as measurements and initializations of physical qubits. Each circuit element is modeled as an ideal operation U_{ideal} followed by an error E with the given probability

$$U_{\text{faulty}} = E \times U_{\text{ideal}}. \tag{19}$$

Furthermore, faulty operators for measurements are placed before the ideal measurement location. We consider depolarizing noise channels on all single- and two-qubit gates, which are described by the noise operators [39,71,72]

$$E_1 \in \{\sigma_k, \forall k \in \{1,2,3\}\} \tag{20}$$

$$E_2 \in \{\sigma_k \otimes \sigma_l, \forall k,l \in \{0,1,2,3\}\}\backslash\{I \otimes I\} \tag{21}$$

with the Pauli matrices $\sigma_k = \{I, X, Y, Z\}$ with $k = 0,1,2,3$. The depolarizing channels are then determined by

$$\epsilon_1(\rho) = (1-p)\rho + \frac{p}{3}\sum_{i=1}^{3} E_1^i \rho E_1^i \tag{22}$$

$$\epsilon_2(\rho) = (1-p)\rho + \frac{p}{15}\sum_{i=1}^{15} E_2^i \rho E_2^i. \tag{23}$$

This means that, one of the 3(15) possible combinations of single(two)-qubit Pauli errors is applied with a given probability $p/3$ $(p/15)$. Qubits are prepared in $|0\rangle$ and measured in the Z basis. We model faults on these state preparations and measurements by applying X errors after state preparations and before measurements each with a probability p.

The logical failure rate is determined as follows. At the end of each MC shot, we perform one round of ideal QEC on the noisy output states, which maps the state back into the codespace. Finally, we extract the expectation value of the corresponding logical operator $\langle\overline{O}\rangle$ classically in software, which indicates that a logical error has occurred if we find an expectation value of -1. Given the expectation value of the logical operator, we can calculate the logical failure rate

$$p_L = \frac{1}{2}\left(1 - \langle\overline{O}\rangle\right). \tag{24}$$

We realize n_{runs} MC shots for each protocol and obtain the logical failure rate p_L by averaging over all shots.

The uncertainties on the logical failure rates are calculated by taking the uncertainty of the mean value for a

binomial distribution

$$\sigma_{p_L} = \sqrt{\frac{p_L(1-p_L)}{n_{\text{runs}}}}, \tag{25}$$

where n_{runs} is the total number of simulation runs and p_L is the estimated logical failure probability. We repeat each simulation 10^5 to 10^6 times, until the relative uncertainty on a given data point is smaller than 5%. Up to 10^6 shots are required for small physical error rates, where error events are rare.

We use the package PECOS, which is a Python framework for studying, developing, and evaluating quantum error-correction protocols through numerically performing stabilizer or state-vector simulations of noisy quantum circuits [73].

VIII. BUILDING BLOCKS USING FT CODE SWITCHING

We determine the logical failure rates for each block given in Fig. 2 by means of the above-described simulation methods. This includes the initialization of the logical states $|\overline{0}\rangle$ and $|\overline{+}\rangle$ on the tetrahedral $[\![15,1,3]\!]$ and the morphed $[\![10,1,2]\!]$ code, FT switching in both directions to and from the $[\![7,1,3]\!]$ Steane code, as well as the composite protocols specified in Fig. 2(c). These protocols implement the Hadamard gate on the tetrahedral code, the T gate on the Steane code and the preparation of a magic state on the Steane code via the morphed code.

For the initialization of logical states in the corresponding codes, we construct the circuits shown in Figs. 20, 21, and 23. The logical failure rates for these protocols are shown in Fig. 10. In the regime of low physical error rates $p \to 0$, we identify a quadratic scaling in the logical failure rate $p_L \sim p^2$ for the FT protocols and a linear scaling $p_L \sim p$ for the non-FT protocol. This indicates that for the non-FT protocol, single errors result in a logical failure, whereas for the FT protocols only weight-2 error configurations contribute to p_L. Since the circuit depth and the number of required gates for the $[\![10,1,2]\!]$ code is smaller, as summarized in Table I, there are fewer weight-2 error configurations that can contribute to the logical failure rate. Thus, the logical failure rates for the initialization of the morphed $[\![10,1,2]\!]$ code are lower than for the tetrahedral code.

Figures 11(a) and 11(b) show the logical failure rates for FT switching with the tetrahedral code. Furthermore, we determine the logical failure rates for non-FT code switching, where the corresponding three stabilizers are measured once without flags. A third simulation considers switching with the tetrahedral code while postselecting for any detected error. Whenever a flag is triggered, or an error on a data qubit is detected, the corresponding run is discarded and the protocol is restarted as discussed in Sec. VI A. For the morphed code, we determine the logical failure rates for FT switching, as well as for the non-FT switching scheme. These protocols are nondeterministic, since the $[\![10,1,2]\!]$ code is an error-detecting code. For non-FT switching, the stabilizers are measured once without flags.

The logical failure rates for FT switching from the tetrahedral $[\![15,1,3]\!]$ code to the $[\![7,1,3]\!]$ Steane code intersects with the logical failure rate for non-FT switching. This crossing point is higher for the inverse direction. The X-error correction block for switching to the Steane code, which is necessary to achieve fault tolerance, requires

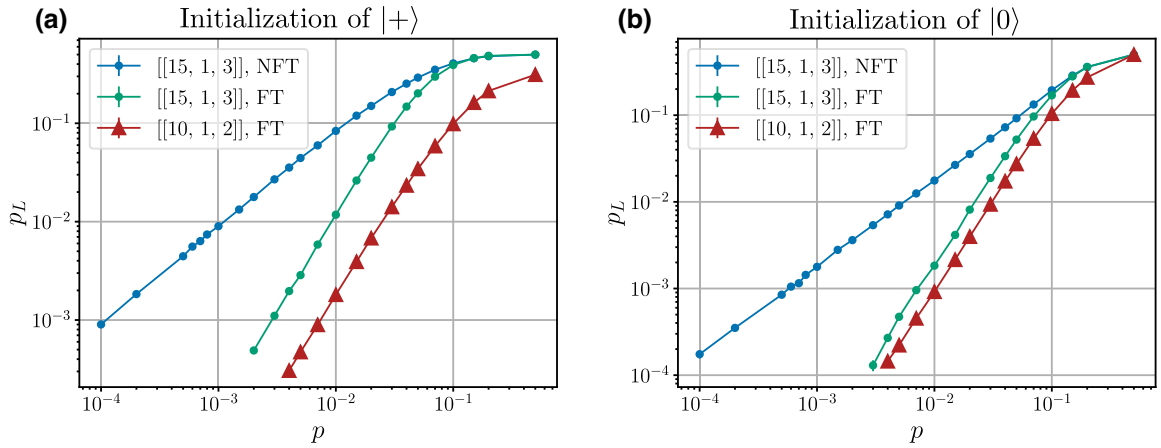

FIG. 10. Logical failure rates for the initialization of logical states illustrated in Fig. 2(b). The logical failure rates are shown for the initialization of (a) $|\overline{0}\rangle$ and (b) $|\overline{+}\rangle$ on the tetrahedral code $[\![15,1,3]\!]$ using a FT scheme that includes a verification step (green) and a non-FT (NFT) scheme without this verification (blue), as well as on the morphed $[\![10,1,2]\!]$ code (dark red). The logical failure rates for the non-FT protocol scale linearly in the error parameter p, since single physical errors can result in a logical failure. For the FT initializations, we can identify a quadratic scaling in p, verifying that only two physical failures introduce a logical error. For the $[\![10,1,2]\!]$ code, lower failure rates are reached than for the initialization of the same state on the tetrahedral code.

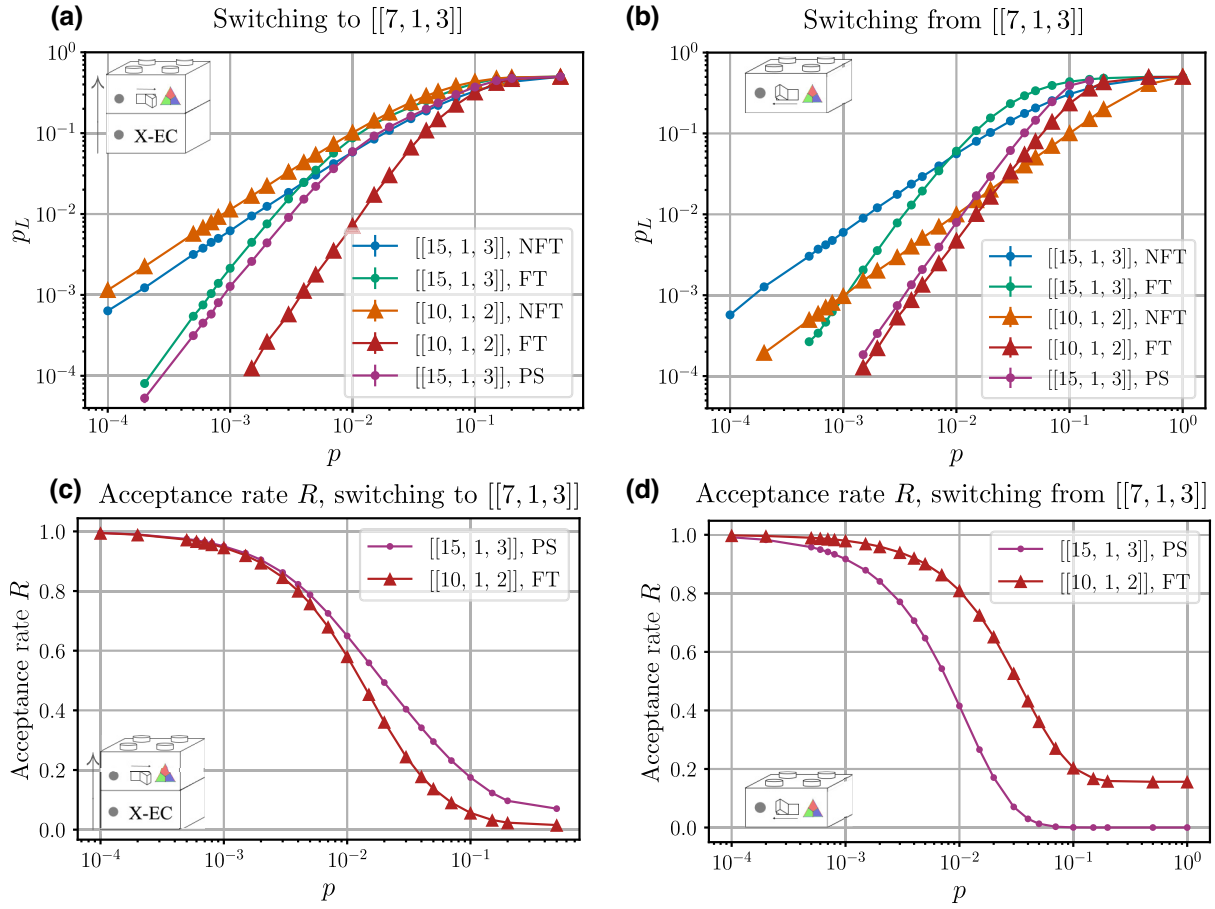

FIG. 11. Logical failure rates and acceptance rates R for building blocks introduced in Fig. 2(b). (a) Logical failure rates for switching to the Steane code from the tetrahedral code using a fault-tolerant (FT, green) scheme, a non-FT (NFT, blue) scheme and postselection (PS, purple), as well as for switching to the Steane code from the morphed code using a FT (dark red) and NFT (orange) scheme. The FT nondeterministic protocols achieve lower logical failure rates than the deterministic ones, while the FT scheme with the $[\![10, 1, 2]\!]$ code performs best. (b) Logical failure rates for switching from the Steane code to the tetrahedral code using a FT (green) scheme, a non-FT (blue) scheme and postselection (purple), as well as for switching from the Steane code to the morphed code using a FT (dark red) and non-FT (orange) scheme. (c) Acceptance rates R for switching from the $[\![15, 1, 3]\!]$ code and the $[\![10, 1, 2]\!]$ code to the Steane code using nondeterministic protocols. The acceptance rates for the $[\![10, 1, 2]\!]$ code are slightly smaller than for the $[\![15, 1, 3]\!]$ code in the considered range of the physical error rate p. (d) Acceptance rates R for the inverse switching direction. The acceptance rates for the $[\![10, 1, 2]\!]$ code are higher than for the $[\![15, 1, 3]\!]$ code in the considered range of the physical error rate p and do not approach 0 for $p \to 1$.

many CNOT gates, as summarized in Table I. This increases the number of possible error configurations and, therefore, leads to an asymmetry between the two switching directions. Considering switching with the tetrahedral $[\![15, 1, 3]\!]$ code, we observe that the non-FT schemes achieve similar values for p_L for both switching directions, while for the FT protocols, these rates differ. The non-FT protocol is symmetric in the switching directions. In both cases, the three weight-4 stabilizers have to be measured and each Pauli error is equally likely in this error model. For the FT-switching protocols, the directions are not symmetric. Similarly, non-FT switching from the $[\![10, 1, 2]\!]$ code to the Steane code achieves higher logical failure rates than the inverse direction, since the stabilizer weights for the two directions differ: for switching to the Steane code, the weight-4 faces are measured, where, for example, a single error on an auxiliary qubit can cause a logical failure. For switching from the Steane code to the $[\![10, 1, 2]\!]$ code, only the weight-3 stabilizers have to be measured.

Figures 11(c) and 11(d) show the acceptance rates for the nondeterministic protocols, which include switching with the tetrahedral code with postselection and FT switching with the morphed code. The acceptance rate

$$R = \frac{n_{\text{runs,accepted}}}{n_{\text{runs,total}}} \tag{26}$$

indicates the number of runs that are not discarded due to an error-detection event divided by the total number of runs. For switching from the $[\![7, 1, 3]\!]$ Steane code to

the morphed $[\![10, 1, 2]\!]$ code, a larger fraction of runs is accepted than for the corresponding protocol with the tetrahedral $[\![15, 1, 3]\!]$ code. This is inverted for the reverse direction, while the relative difference between the acceptance rates decreases. Furthermore, the acceptance rates do not go to 0 as $p \to 1$. One reason for this is that any error configuration that directly corresponds to a logical operator, is not detected and, therefore, there is always a certain fraction of runs that is not discarded and causes a logical failure. The finite acceptance rate for $p \to 1$ considering FT switching from $[\![7, 1, 3]\!]$ to $[\![10, 1, 2]\!]$ can be traced back to the partial error detection that is happening on the target $[\![10, 1, 2]\!]$ code: some Z errors are only detectable and we have to discard the corresponding run if the respective syndrome is measured, while other Z errors are still correctable. If we postselect for any detected Z error on the $[\![10, 1, 2]\!]$ in the end, we find that the acceptance rate converges to similar values as for the tetrahedral code, which is discussed in more detail in Appendix B. Accordingly, note that for lower p values, this also leads to lower logical failure rates p_L, at the expense of a further reduced acceptance rate.

Figures 12(a) and 12(b) show the logical failure and acceptance rates for the gates that cannot be realized transversally in the corresponding codes but are necessary in order to complete the universal gate set. For the Hadamard gate on the tetrahedral $[\![15, 1, 3]\!]$ code, first, switching to the $[\![7, 1, 3]\!]$ Steane code is applied, followed by a transversal Hadamard operation on the Steane code and switching back to the tetrahedral code. Analogously,

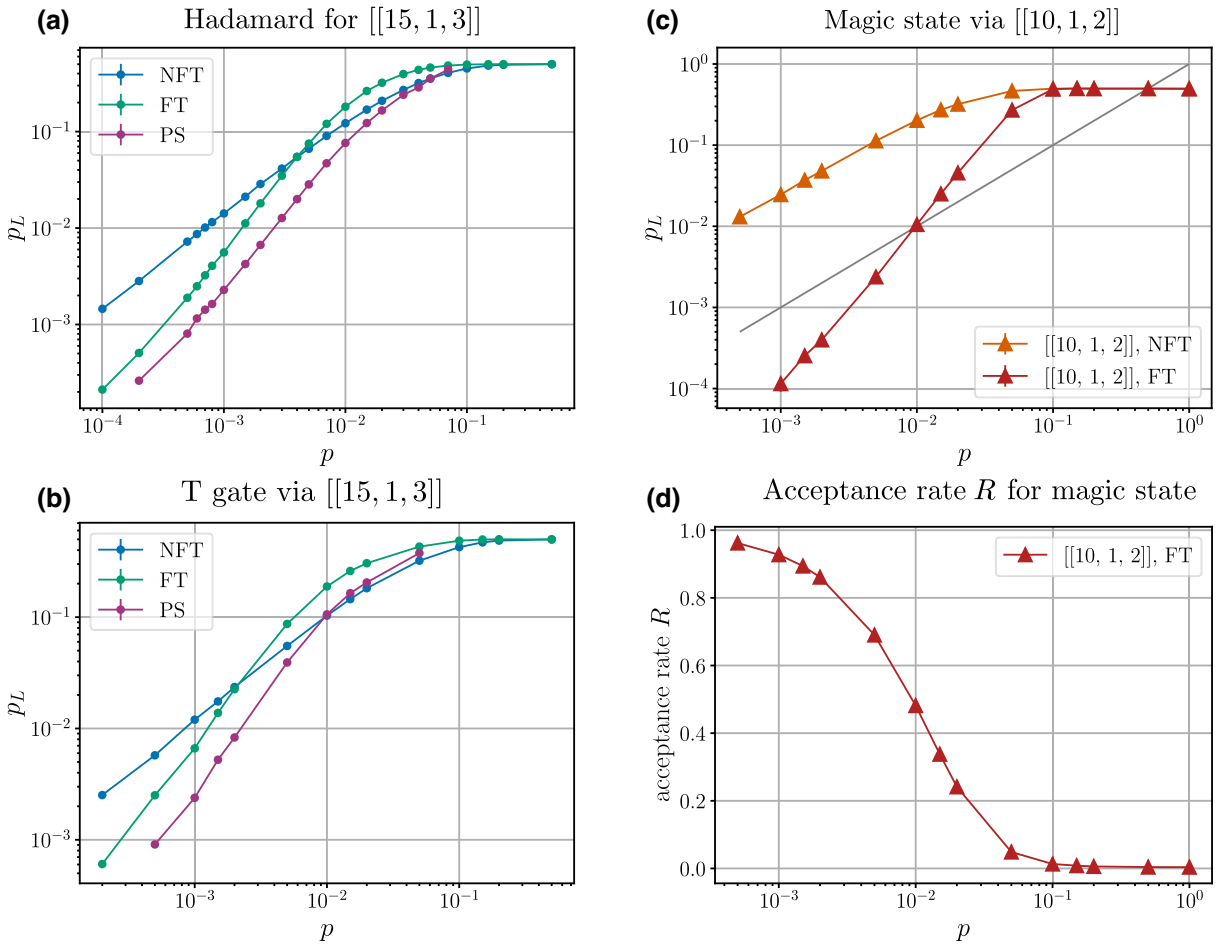

FIG. 12. Logical failure rates for logical operations illustrated in Fig. 2(c). The logical failure rates are averaged over different initial states for the non-FT (blue), FT (green), and postselected (PS, purple) implementation of (a) the Hadamard gate on the tetrahedral $[\![15, 1, 3]\!]$ code and (b) the T gate on the Steane code. The non-FT implementations of both gates achieve similar logical failure rates, while the FT versions perform worse for the T gate than for the Hadamard gate. Note that for a specific input state, these failure rates may look different. Different states are sensitive to different kinds of errors and the amount of dangerous error positions throughout the protocols varies for each type of error. (c) Logical failure rates for the preparation of the magic state $\overline{\top}|\overline{+}\rangle$ on the $[\![7, 1, 3]\!]$ Steane code using the morphed $[\![10, 1, 2]\!]$ code for the non-FT (orange) and FT (red) scheme. The gray line corresponds to the physical error rate p. For $p < 10^{-2}$, the FT scheme the logical failure rate is smaller than p. (d) Acceptance rates R for magic state preparation with the morphed $[\![10, 1, 2]\!]$ code. For $p < 10^{-2}$, more than half of the runs is accepted.

the T gate on the Steane code is implemented by switching to the tetrahedral code, applying the transversal T gate and switching back to the initial code. Again, we consider FT protocols, as discussed in the previous sections, as well as non-FT switching and a nondeterministic version of the corresponding protocol, which includes postselection for any detected error.

The FT protocols outperform only the non-FT schemes for $p \leq 5 \times 10^{-3}$, due to the large overhead that is required to achieve fault tolerance. Again, we observe that the failure rates for the non-FT scheme on the tetrahedral code are similar for both operations. In this case, the switching cycles are symmetric, meaning that after the first switching step in one protocol, the probability for a given number of errors is the same as after the first switching step for the other protocol. For the FT protocols, these may differ, since the two switching directions include different operations and a different number of stabilizer measurements. Non-FT switching from the Steane code to the morphed $[\![10, 1, 2]\!]$ code achieves lower logical failure rates than for the inverse direction. For switching from the Steane code, we need only to measure the weight-3 Z stabilizers. Any error on the auxiliary qubits may propagate but is always equivalent to a weight-1 error. For switching in the inverse direction, the weight-4 faces have to be measured requiring more two-qubit gates and giving rise to logical failures due to propagated errors on the auxiliary qubits.

Figures 12(c) and 12(d) show the logical failure rates and acceptance rates for the preparation of a magic state on the Steane code using the morphed $[\![10, 1, 2]\!]$ code. The logical failure rates scale quadratically, indicating that no single error results in a logical failure. The logical failure rate of the FT scheme surpasses the corresponding physical error rate at approximately $p = 10^{-2}$, while keeping more than 50% of the runs below this point.

To summarize, we observe the characteristic quadratic scaling at low physical error rates for the constructed FT code-switching protocols. In addition, we find that the non-FT code-switching protocol for switching from the $[\![7, 1, 3]\!]$ Steane code to the morphed $[\![10, 1, 2]\!]$ code achieves lower physical error rates than FT switching with the tetrahedral $[\![15, 1, 3]\!]$ code for $p \geq 10^{-3}$. Furthermore, we estimate a breakeven point at a physical error rate of approximately 1×10^{-2} with an acceptance rate of 50% for the preparation of a magic state on the Steane code using the morphed $[\![10, 1, 2]\!]$ code.

IX. PROJECTED PERFORMANCE FOR TRAPPED-ION QUANTUM PROCESSORS

Noise on real devices cannot be described by a single-parameter noise model. Error rates can vary for different circuit components, for example, the two-qubit gate error rates are typically higher than those of other error sources. In order to estimate the projected performance on near-term trapped-ion quantum processors, we consider a multiparameter noise model with different error rates for each type of circuit component [25,39,74,75].

We simulate a Hadamard gate for the tetrahedral code, a T gate for the Steane code and the preparation of a magic state on the Steane code using a modified noise model. We specify the depolarizing channel on single-qubit gates with a parameter p_1 and on two-qubit gates with p_2. Faulty measurements and initializations are modeled with faults with a probability p_m and p_i, respectively, with which the state of the qubit is inverted. In the following simulations, we choose parameter values based on recent benchmarks on ion-trap processors [19] as $p_1 = 1 \times 10^{-4}, p_2 = 3 \times 10^{-3}, p_i = 1 \times 10^{-3}$, and $p_m = 1 \times 10^{-3}$. A detailed discussion on the modeling of noise in ion traps [19,71,72] is beyond the scope of this work.

Figure 13(a) shows the averaged logical failure rates for deterministic gates using code-switching protocols. Both logical failure rates for the deterministic Hadamard gate on the tetrahedral $[\![15, 1, 3]\!]$ code and the deterministic T gate on the Steane code are much larger than the estimated corresponding physical error rate, but are comparable to recent benchmarks on trapped-ion setups [19,39,41].

Figure 13(b) shows the averaged logical failure rates for magic state preparation on the Steane code for three different protocols. The left bar corresponds to the initialization of $|\overline{+}\rangle$ on the $[\![15, 1, 3]\!]$ tetrahedral code, the application of a T gate and, then, switching to the Steane code while postselecting for errors on data and auxiliary qubits. Using the morphed $[\![10, 1, 2]\!]$ code for the same protocol yields a logical failure rate, which is below the corresponding failure probability of a single physical qubit. Using the morphed $[\![10, 1, 2]\!]$ code for the preparation of a magic state reduces p_L by 2 orders of magnitude compared to the postselected implementation for the tetrahedral $[\![15, 1, 3]\!]$ code.

In order to compare our results to existing methods, we simulate the preparation of a magic state on the Steane code using a state-of-the-art method [39–41]. Here, the magic state is prepared non-fault-tolerantly on the Steane code, followed by a measurement of a logical operator and one round of error detection on the Steane code. We observe that the logical failure rate for magic state preparation with the morphed $[\![10, 1, 2]\!]$ code is competitive to state-of-the-art magic state preparation on the Steane code [39,65] in terms of the achievable fidelity, while the protocol starting with the morphed code requires fewer two-qubit gates.

The logical failure rates shown in Fig. 13 are averaged over different initial input states. However, the number of Pauli-X- and Z-error configurations that cause a logical failure may vary, depending on the direction of switching and the given input state. Figure 14(a) shows the failure rates for the deterministic Hadamard and T gate for ideal logical input states $|\overline{0}\rangle$ and $|\overline{+}\rangle$. We observe that the failure

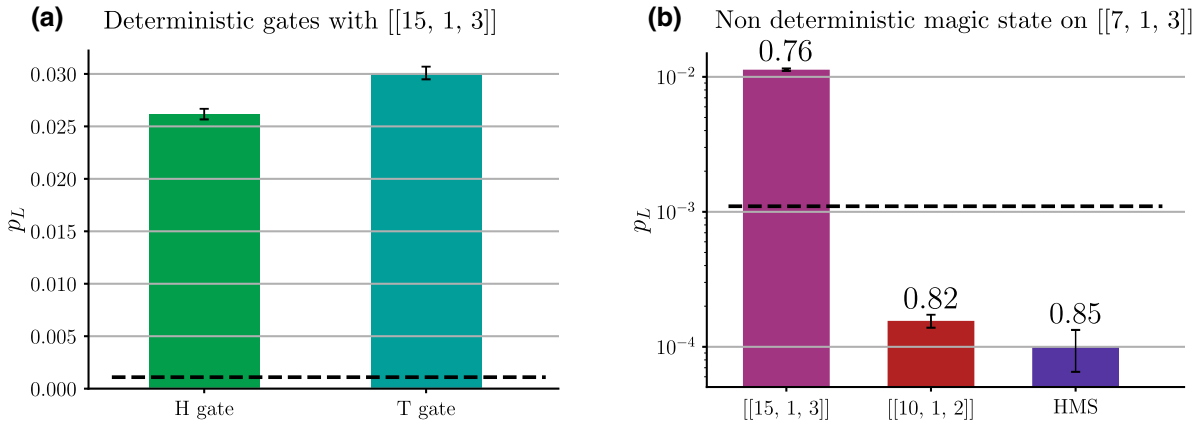

FIG. 13. Projected performance of logical operations on trapped-ion quantum processors. (a) The projected performance for deterministic Hadamard (left) and T gate (right) is estimated for current ion-trap setups [25]. These protocols correspond to the two left composite columns in Fig. 2(c). The shown logical failure rates are calculated for an ideal input state. The black dashed line corresponds to the failure rate of the according sequence for a single physical qubit. For both logical gates, the determined logical failure rate on near-term ion-trap processors is still more than one order of magnitude larger than the expected rate for a single physical qubit. (b) Logical failure rates for the preparation of the magic state $\overline{\mathrm{T}}|\overline{+}\rangle$ on the Steane code on a logarithmic scale. These protocols correspond to the composite column in Fig. 2(c) on the right, including a noisy initialization, a noisy application of the T gate, and noisy switching to the Steane code. For the tetrahedral $[\![15, 1, 3]\!]$ code (left) with postselection, we obtain logical failure rates which are clearly above the breakeven point (black dashed line) at an acceptance rate of $R = 76\%$. Using the morphed $[\![10, 1, 2]\!]$ code (center) lowers p_L by 2 orders of magnitude and increased the acceptance rate to $R = 82\%$. *HMS* corresponds to the preparation of a heralded magic state on the Steane code using a state-of-the-art FT implementation [39]. For this scheme, we obtain the lowest failure rates at an acceptance rate of $R = 85\%$.

rates for the logical input state $|\overline{+}\rangle$ are higher than for the input state $|\overline{0}\rangle$. Considering the Hadamard gate, this can be explained by looking at the dangerous Pauli-error positions on data qubits. As discussed in Sec. V, Pauli-Z errors on data qubits can induce a logical failure when switching from the tetrahedral $[\![15, 1, 3]\!]$ code to the $[\![7, 1, 3]\!]$ Steane code. Analogously, Pauli-X errors on data qubits can cause a logical failure when switching from the Steane code to the tetrahedral code. For the initial state $|\overline{0}\rangle$, Pauli-Z errors do not affect the encoded state, since $Z|0\rangle = |0\rangle$. After switching to the Steane code, the state is changed to $|\overline{+}\rangle$ by applying the Hadamard gate on the Steane code. The dangerous Pauli-X errors during the second switching step do not affect this encoded state. So, the Hadamard gate on $|\overline{0}\rangle$ on the tetrahedral code is inherently less sensitive to these errors on data qubits than on the initial state

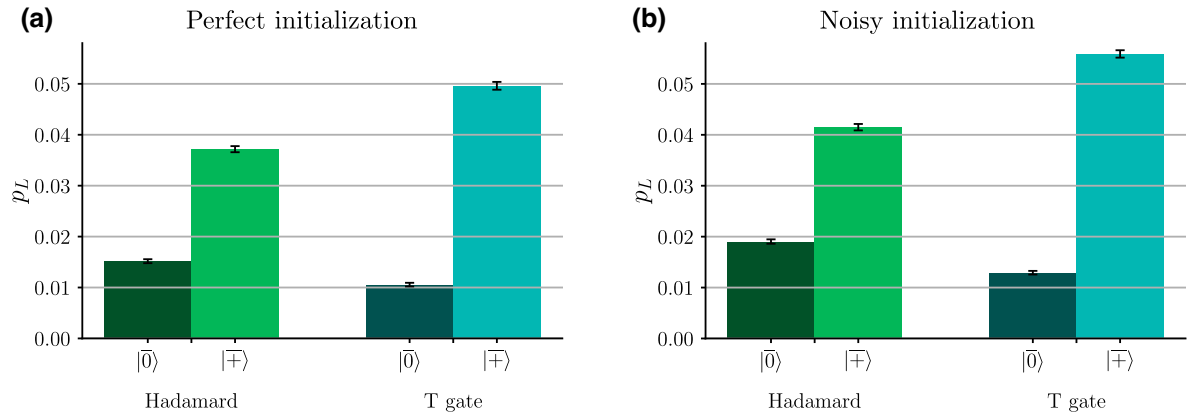

FIG. 14. Projected performance of deterministic logical gates for different encoded states. Projected performance for deterministic Hadamard (left) and T gate (right) for current ion-trap setups [25] for logical input states $|\overline{0}\rangle$ and $|\overline{+}\rangle$ (a) for perfect and (b) noisy initialization. The Hadamard on a perfect state $|\overline{0}\rangle$ achieves lower logical failure rates than on a perfect state $|\overline{+}\rangle$. Similarly for the T gate, we find higher logical failure rates for $|\overline{+}\rangle$ than for $|\overline{0}\rangle$. With the noisy initialization, all logical failure rates increase slightly while the systematic difference between $|\overline{0}\rangle$ and $|\overline{+}\rangle$ is maintained.

TABLE II. Projected logical failure rates for logical operations on an ion-trap quantum processor. For the noise parameters specified in Sec. IX, which are based on current ion-trap setups, we determined the logical failure rates for logical operations using FT code-switching protocols.

Logical operation	Code	Figure	$p_L \times 10^{-2}$
H	$[\![15,1,3]\!]$	13(a)	2.62(5)
T	$[\![7,1,3]\!]$	13(a)	3.01(6)
Magic state	$[\![15,1,3]\!]$	13(b)	1.13(2)
Magic state	$[\![10,1,2]\!]$	13(b)	$1.60(2) \times 10^{-2}$
Heralded magic state		13(b)	$1.00(3) \times 10^{-2}$
H for perfect $\|\bar{0}\rangle$	$[\![15,1,3]\!]$	14(a)	1.52(4)
H for perfect $\|\bar{+}\rangle$	$[\![15,1,3]\!]$	14(a)	3.72(6)
T for perfect $\|\bar{0}\rangle$	$[\![15,1,3]\!]$	14(a)	1.05(4)
T for perfect $\|\bar{+}\rangle$	$[\![15,1,3]\!]$	14(a)	4.96(8)
H for noisy $\|\bar{0}\rangle$	$[\![15,1,3]\!]$	14(b)	1.90(4)
H for noisy $\|\bar{+}\rangle$	$[\![15,1,3]\!]$	14(b)	4.15(6)
T for noisy $\|\bar{0}\rangle$	$[\![15,1,3]\!]$	14(b)	1.29(4)
T for noisy $\|\bar{+}\rangle$	$[\![15,1,3]\!]$	14(b)	5.59(7)

$|\bar{+}\rangle$. Furthermore, $|\bar{0}\rangle$ is an eigenstate of the T gate, so it is expected that logical failure rates are lower than for other states, in agreement with recent experimental observations [39]. The values of the numerically obtained logical failure rates presented in Figs. 13 and 14 are summarized in Table II.

To estimate the impact of an imperfect initialization, we determine the logical failure rates for the same protocols including a noisy initialization which is shown in Fig. 14(b). The initializations are implemented using the circuits shown in Figs. 20–24. Including a noisy initialization increases p_L by a factor of approximately 1.1. This indicates that the logical failure rate is dominated by the noisy switching procedure, rather than by the noisy initialization of logical input states.

We identify the parameters p_2 as a limiting factor for the above code-switching schemes. It is the largest value in our noise model and the number of two-qubit gates is much higher than for any other component. In order to estimate the sensitivity of the considered protocols to changes in p_2, we choose $p_1 = p_i = p_m = 0$ and only vary the two-qubit gate parameter p_2. Figure 15 shows the averaged logical failure rates for the composite protocols as specified in Fig. 2(c) for this noise model. The logical failure rate for the initial value of $p_2 = 3 \times 10^{-3}$ for the Hadamard gate is close to the logical failure rates for the above noise model, where we included errors on single-qubit operations. This is in alignment with the expectation that faulty single-qubit operations have a minor impact on the logical failure rate and the probability for errors on two-qubit gates p_2 is the limiting parameter. Furthermore, we observe that an improvement of a factor of 5 should suffice to reach the regime of the corresponding physical error rate for a FT nondeterministic magic state preparation, as shown in Fig. 15(b) on the left.

Overall, we report that the morphed $[\![10,1,2]\!]$ code can reduce the logical failure rates by up to 2 orders of magnitude compared to the tetrahedral $[\![15,1,3]\!]$ code, as, for example, for magic state preparation shown in Fig. 13.

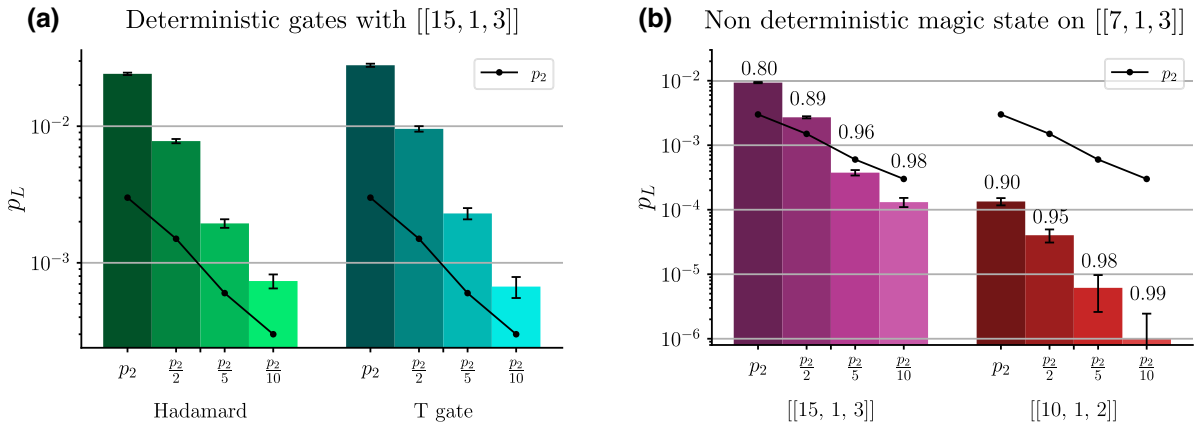

FIG. 15. Scaling of logical failure rates with the two-qubit gate noise parameter p_2 for different protocols. (a) Averaged logical failure rates for the deterministic Hadamard gate on the $[\![15,1,3]\!]$ tetrahedral code and the deterministic T gate on the Steane code according to the protocol specified in Fig. 2(c) given an ideal logical input state. The parameters p_1, p_i, and p_m are set to 0 and p_2 is reduced by a factor n for $n = 1, 2, 5, 10$. The black dots correspond to the physical error rate of p_2/n. The logical failure rates are larger than this physical error rate for the considered variation in p_2. (b) Averaged logical failure rates for the preparation of a magic state on the Steane code on a logarithmic scale as specified in Fig. 2(c). The preparation of the magic state includes the noisy initialization of the $[\![15,1,3]\!]$ (left) and $[\![10,1,2]\!]$ (right) code, the application of the FT T gate and switching to the $[\![7,1,3]\!]$ Steane code. We consider the same variation in the parameter p_2 for $p_1 = p_i = p_m = 0$. For magic state preparation using the tetrahedral $[\![15,1,3]\!]$ code and postselection, the logical failure rate succeeds the corresponding physical failure rate at $p_2/5$, while for the morphed $[\![10,1,2]\!]$ code, p_L is already below this rate for the initial value of p_2.

Furthermore, the FT deterministic logical gates are within reach of recently proposed implementations [39,41,65].

X. CONCLUSIONS AND OUTLOOK

In this work, we have constructed fault-tolerant code-switching protocols that enable the implementation of a FT universal gate set on the logical level. We provide specific instructions for the implementation of code-switching protocols based on FT building blocks offering multiple methods as summarized in Fig. 2. Composite code-switching protocols can be constructed from this suite by composing building blocks in ways that are most suitable and optimized for given experimental setups. Based on current ion-trap quantum processors, we estimate the performance of code-switching protocols on near-term devices using a multiparameter noise model and find that the logical failure rates reach values on par with state-of-the-art magic state injection [25,39].

A study comparing the overhead and thresholds for preparing magic states on the Steane code using magic state distillation (MSD) and deterministic code switching does not find an advantage in using code switching over MSD for this task [65]. We complement this study by considering morphed codes. As demonstrated in our work, stabilizer weights and, therefore, the circuit depth of a given protocol can be reduced by replacing the distance-three tetrahedral color code with the respective morphed code, which offers a feasible alternative for magic state preparation on near-term devices.

The presented strategies for FT code switching can be extended to higher distances. The general scheme of code switching is not limited to low code distances [44,61,65,76]. A fault-tolerant switching protocol will require (repeated) measurement of higher-weight stabilizers, for which techniques to construct flag-qubit-based fault-tolerant stabilizer measurement circuits have been worked out for arbitrary distance codes [52]. Furthermore, general tetrahedral color codes can be morphed into the class of *hybrid color-toric* codes [55], which can be used analogously to the described methods for switching with the respective two-dimensional color code. By additionally increasing the number of repeated stabilizer measurements, as specified by the code's distance, one can, in principle, generalize the approach explored in this work to achieve FT code switching for larger-distance codes.

Our numerical results provide a basis for the experimental realization of code-switching protocols on existing or near-term quantum processors. Our protocols are readily implementable in architectures offering all-to-all qubit connectivity, such as, e.g., in ion-trap quantum processors [77,78]. It would be interesting to adapt the protocols and also consider connectivity of other platforms, in particular, nearest-neighbor connectivity of superconducting qubits [79] or dynamically reconfigurable quantum registers of neutral atom processors [32,80] or shuttling-based trapped-ion architectures [77,81]. We emphasize that the proposed protocols and quantum circuit constructions are not limited to low-distance codes, but can be readily leveraged to scalable larger-distance two- and three-dimensional color codes. Furthermore, tailoring FT code switching to setups with biased noise [82–84] offers the potential of reaching even lower logical failure rates for suitable experimental systems with such noise characteristics.

ACKNOWLEDGMENTS

We would like to thank Josias Old for valuable discussions on FT-encoding circuits. We gratefully acknowledge support by the EU Quantum Technology Flagship Grant under Grant Agreement No. 820495 (AQTION), the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101114305 ("MILLENION-SGA1" EU Project), the U.S. Army Research Office through Grant No. W911NF-21-1-0007, the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101046968 (BRISQ), the ERC Starting Grant QNets through Grant No. 804247, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy "Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1" 390534769. This research is also part of the Munich Quantum Valley (K-8), which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. This research is also supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office through Grant No. W911NF-16-1-0070. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the U.S. Army Research Office. The numerical simulations were performed with the aid of computing resources at Forschungszentrum Jülich.

APPENDIX A: SUMMARY FT CODE-SWITCHING PROTOCOLS

The following two algorithms summarize the protocols that we implement for FT switching between $[\![15, 1, 3]\!]$ and $[\![7, 1, 3]\!]$, which is described in detail in Sec. V. These include the repetition of measurements in order to correct any single measurement fault, using the flag qubit scheme

Protocol . FT switching $[\![15,1,3]\!] \rightarrow [\![7,1,3]\!]$

Input: Logical state in $[[15,1,3]]$
Output: Logical state in $[[7,1,3]]$
Measure (S_R^X, S_B^X, S_G^X) with flags
  if a circuit flagged:
    Remeasure (S_R^X, S_B^X, S_G^X) without flags
    Measure one additional stabilizer
    correct flag errors
Measure the bulk (yellow cell) destructively
Check agreement of opposing X-faces
  if $B_Y^X == -1$ and agreement $\neq (0, 0, 0)$:
    $(S_R^X, S_B^X, S_G^X) \longrightarrow (S_R^X, S_B^X, S_G^X)$+ agreement %2
Apply switching operation according to lookup table

Protocol . FT switching $[\![7,1,3]\!] \rightarrow [\![15,1,3]\!]$

Input: Logical state in $[[7,1,3]]$
Output: Logical state in $[[15,1,3]]$
Measure $(B_{RB}^Z, B_{RG}^Z, B_{BG}^Z)$ twice with flags
  if a circuit flagged:
    Remeasure $(B_{RB}^Z, B_{RG}^Z, B_{BG}^Z)$ without flags
    Measure B_Y^X
    Correct flag errors
  if syndromes disagree:
    Remeasure $(B_{RB}^Z, B_{RG}^Z, B_{BG}^Z)$ without flags
Measure Steane faces (S_R^Z, S_R^Z, S_G^Z) twice with flags
  if a circuit flagged:
    Remeasure (S_R^Z, S_R^Z, S_G^Z) without flags
    Measure one additional stabilizer
    Correct flag errors
  if syndromes disagree:
    Remeasure $(S_R^Z S_R^Z, S_G^Z)$ without flags
Update syndrome based on detected error
Apply switching operation according to lookup table

to correct for fault on auxiliary qubits as well as the performance of additional stabilizer measurements to correct for potentially dangerous errors on data qubits.

APPENDIX B: MORPHING THE TETRAHEDRAL $[\![15,1,3]\!]$ CODE INTO THE $[\![10,1,2]\!]$ CODE

The parent $[\![15,1,3]\!]$ code defined on the tetrahedron contains a smaller subcode on the yellow cell, which is a $[\![8,3,2]\!]$ stabilizer code. The stabilizers of the $[\![8,3,2]\!]$ code are [69]

$$\begin{aligned} S_X &= X_0X_1X_2X_3X_7X_{12}X_{13}X_{14}, \\ S_Z^1 &= Z_0Z_1Z_2Z_3Z_7Z_{12}Z_{13}Z_{14}, \\ S_Z^2 &= Z_0Z_2Z_{12}Z_{14}, \\ S_Z^3 &= Z_1Z_7Z_{12}Z_{14}, \\ S_Z^4 &= Z_0Z_3Z_7Z_{14}. \end{aligned} \tag{B1}$$

for the indexing given in Fig. 16(b). The logical operators are given by

$$\begin{aligned} \overline{X_1} &= X_0X_2X_3X_{13}, \quad \overline{Z_1} = Z_0Z_{14} \\ \overline{X_2} &= X_1X_3X_7X_{13}, \quad \overline{Z_2} = Z_7Z_{14} \\ \overline{X_3} &= X_1X_2X_{12}X_{13}, \quad \overline{Z_3} = Z_{12}Z_{14}. \end{aligned} \tag{B2}$$

A logical state $|\overline{\psi_1}\,\overline{\psi_2}\,\overline{\psi_3}\rangle$ can be encoded using the circuit shown in Fig. 16. We invert this encoding circuit by reversing the ordering of the applied operations and the direction of all CNOT gates. We apply this inverted circuit

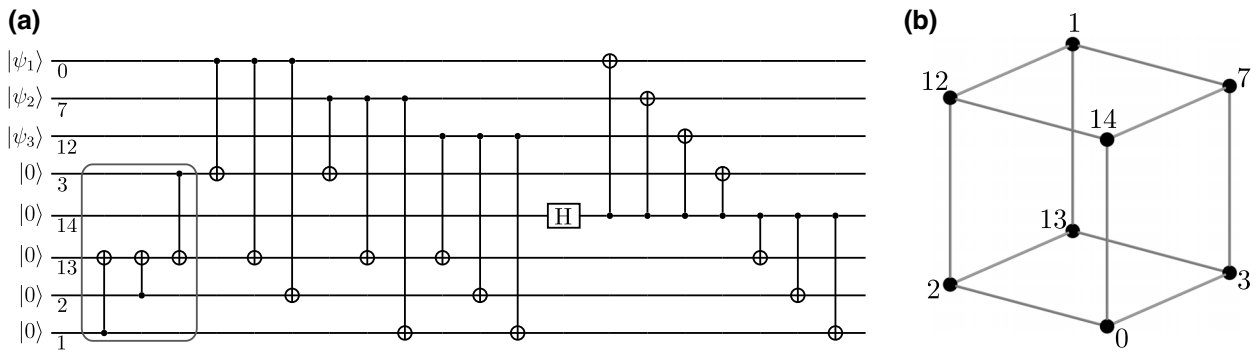

FIG. 16. Encoding of a logical state $|\overline{\psi_1}\,\overline{\psi_2}\,\overline{\psi_3}\rangle$ on the $[\![8,3,2]\!]$ code. We construct this circuit by initializing a $+1$ eigenstate of all stabilizers, as given in Eq. (B1) and coupling qubits 0, 1, and 2, which are in state $|\psi_1\rangle$, $|\psi_2\rangle$, and $|\psi_3\rangle$, to the qubits of the corresponding logical operator. The first three CNOT gates are not required for the encoding of the $[\![8,3,2]\!]$ code, however they are necessary when using the inverted circuit for morphing the tetrahedral code in order to obtain the desired stabilizer operators.

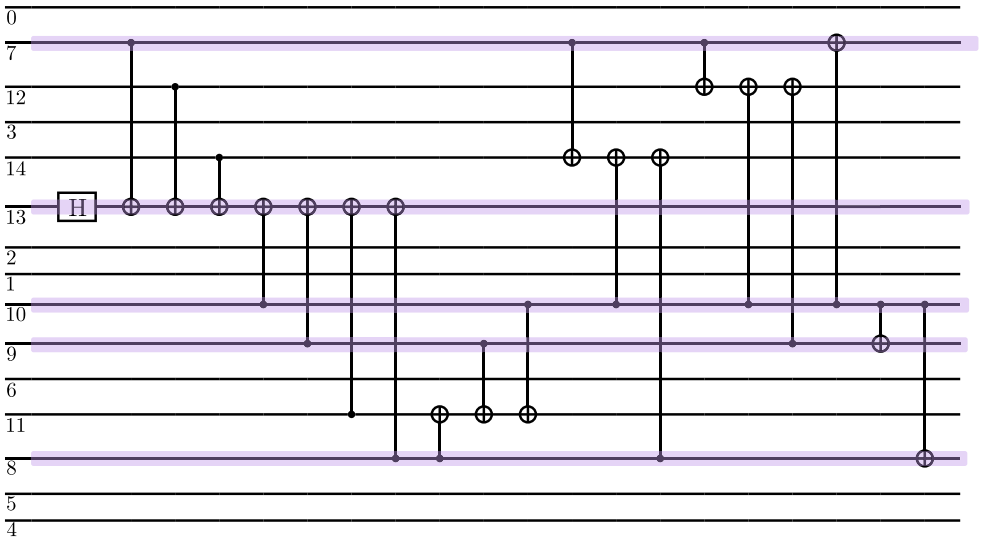

FIG. 17. Circuit for morphing the $[\![15,1,3]\!]$ code into the $[\![10,1,2]\!]$. We find this circuit by inverting the encoding circuit of the child $[\![8,3,2]\!]$ code and applying it to the yellow cell of the tetrahedral code for the indexing specified in Fig. 3(b). The marked qubits are effectively decoupled from the remaining ten qubits.

to a codestate of the $[\![15,1,3]\!]$ parent code, as shown in Fig. 17. In doing so, we effectively decouple five qubits, meaning that they are not entangled with the remaining ten qubits, while the initial information is still encoded on these ten remaining qubits. It is important to note that the circuit shown in Fig. 17 is not physically executed, since it is meant to generate a new code out of an existing one. For our protocols, we use the obtained morphed $[\![10,1,2]\!]$ code directly.

Figure 18 summarizes the possible correctable and detectable Z-error configurations with their respective syndromes for the morphed $[\![10,1,2]\!]$ code. Figure 19 shows the comparison between this partial ED on the target code after switching from the Steane code to the morphed $[\![10,1,2]\!]$ code and full ED, where we discard a run whenever any Z error is detected. The logical failure rate decreases when we postselect for any detected Z error, at the expense of a further lowering of the acceptance rate.

Error	syndrome $(B_R'^X, B_G'^X, B_B'^X)$	correction
Z_0	(100)	discard
Z_1	(110)	Z_1
Z_2	(111)	Z_2
Z_3	(101)	Z_3
Z_4	(010)	discard
Z_5	(011)	Z_5
Z_6	(001)	discard
Z_7	(100)	discard
Z_8	(010)	discard
Z_9	(001)	discard

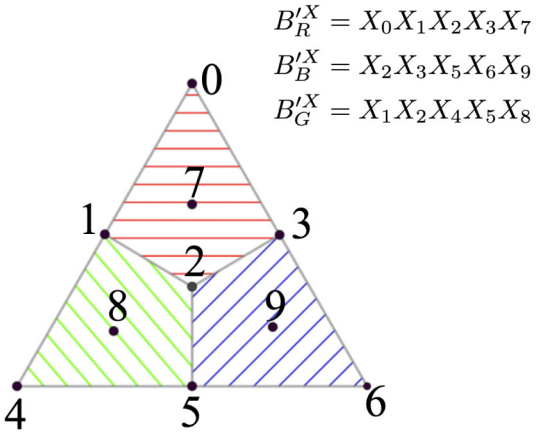

FIG. 18. Lookup table for Z errors on the $[\![10,1,2]\!]$ code. Single Z-error configurations, the resulting syndrome measurement for the above-defined stabilizers and the correction that needs to be applied to get back into the codespace for the qubit indices as specified in Fig. 8. For example, Z_0 and Z_7 are not distinguishable, because they are both part of only one cell. Therefore, it is not possible to correct for all single Z errors.

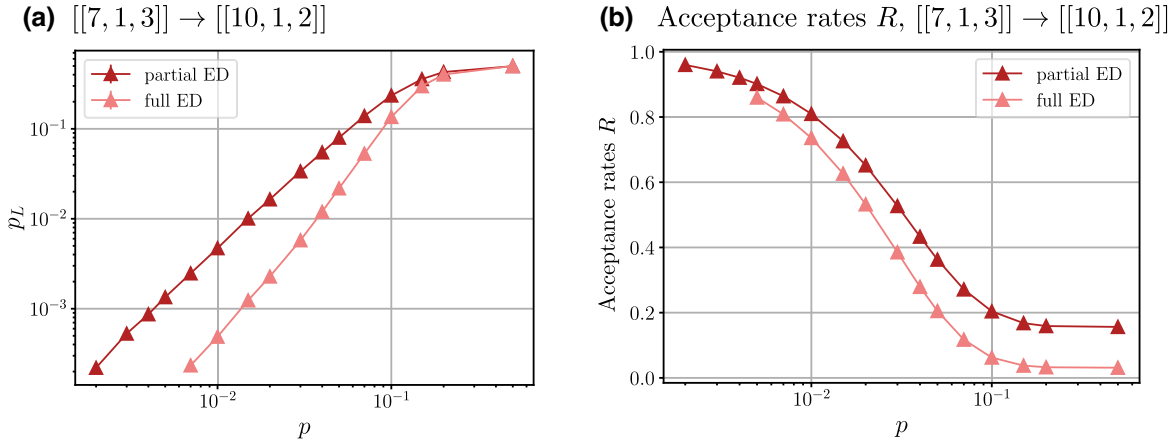

FIG. 19. Logical failure rates and acceptance rates for partial and full ED. (a) Logical failure rates for switching from the Steane code to the morphed $[\![10, 1, 2]\!]$ code for two different versions of error detection. In partial ED, we postselect only on those errors which we cannot distinguish from other weight-1 error configurations, as given in Table 18. For full ED, we discard the corresponding run whenever any Z error is detected in the final round of classical EC on the target code. (b) Acceptance rates for the two versions of ED. In total, we perform up to 10^6 Monte Carlo shots for each value of p. Out of these, fewer runs are accepted for full ED.

APPENDIX C: ENCODING CIRCUITS

The following circuits shown in Figs. 20–24 can be used to implement logical states $\overline{|0\rangle}$ and $\overline{|+\rangle}$ on the tetrahedral code $[\![15, 1, 3]\!]$ and the morphed $[\![10, 1, 2]\!]$ code. We obtain the circuits using the Latin rectangle method [85].

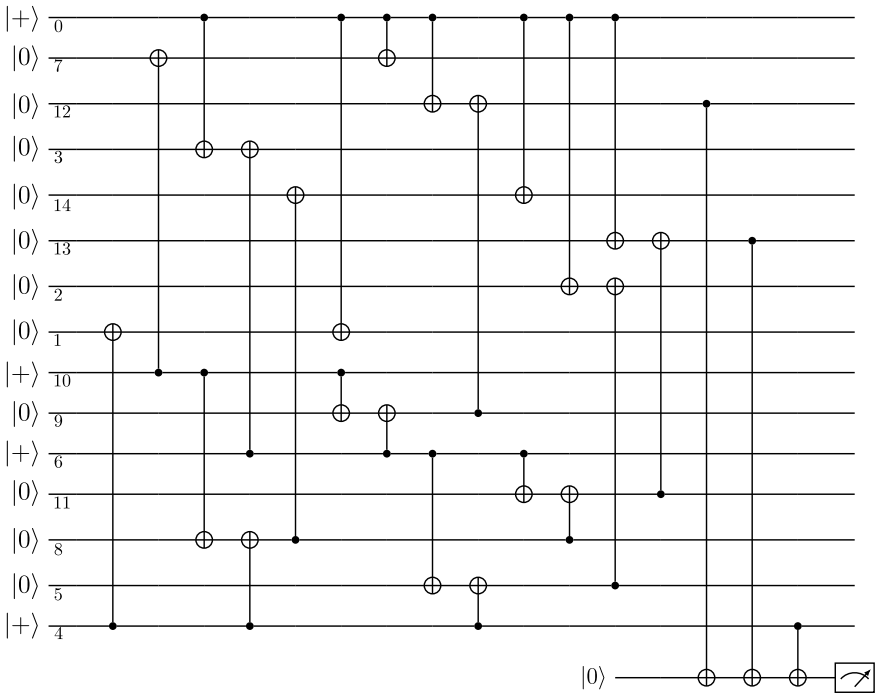

FIG. 20. Circuit for the initialization of $|\overline{0}\rangle$ on the 15-qubit tetrahedral code $[\![15, 1, 3]\!]$. We obtain this circuit using the Latin rectangle method [85]. A verification step is added in order to achieve fault tolerance. This verification corresponds to the measurement of a logical $\overline{Z} = Z_2Z_5Z_{14}$ and detects all incorrectable weight-4 X errors that result from a single propagated error. In total, 25 CNOT gates and 16 qubits are required to fault tolerantly initialize $|\overline{0}\rangle$.

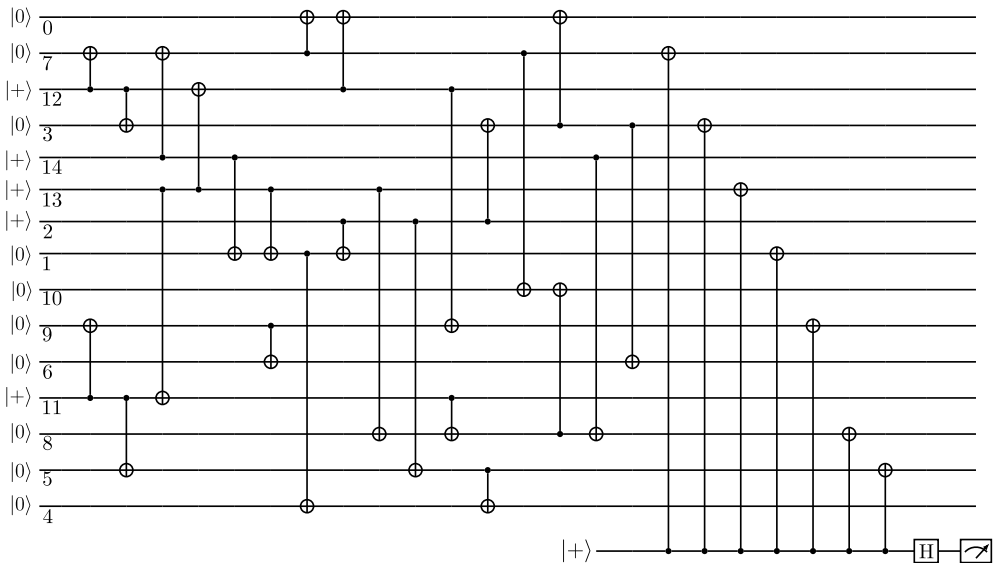

FIG. 21. Circuit for the initialization of $|\overline{+}\rangle$ on the 15-qubit tetrahedral code $[\![15, 1, 3]\!]$. We construct this circuit using the Latin rectangle method [85]. In this case, the verification step corresponds to the measurement of a logical $\overline{X} = X_1X_3X_5X_7X_9X_{12}X_{13}$ and it detects all incorrectable weight-2 Z errors that result from a single propagated error. In total, 32 CNOT gates and 16 qubits are required to fault tolerantly initialize $|\overline{+}\rangle$.

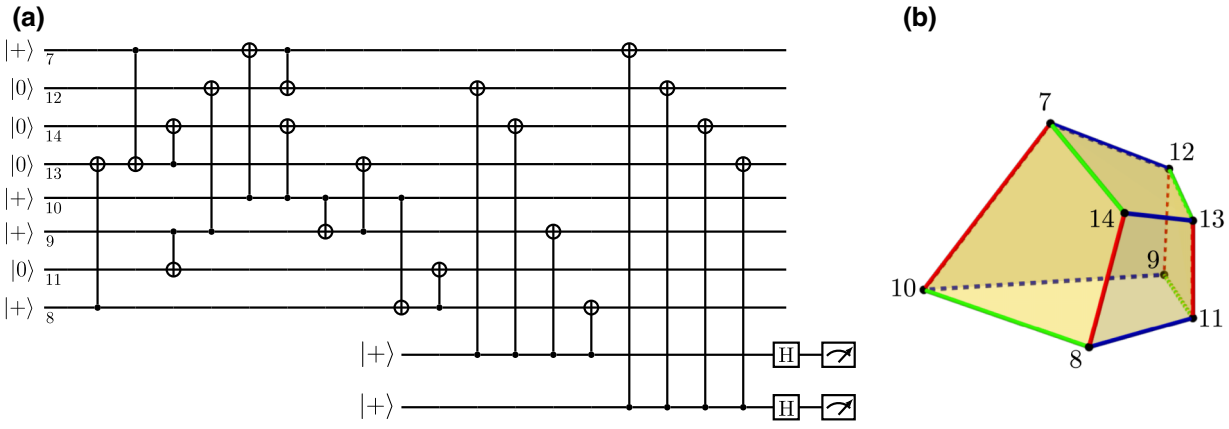

FIG. 22. Circuit for preparing the bulk for code switching. (a) Encoding circuit of the initialization of the bulk (yellow cell) for switching from the Steane code to the tetrahedral $[\![15, 1, 3]\!]$ code. Since the bulk has to be prepared in a specific state but does not encode any information, there is no logical operator that can be used in a verification step. Therefore, two stabilizers have to be measured in order to verify the resulting state. We choose the stabilizers $X_1X_4X_8X_{12}$ and $X_1X_2X_4X_5$ in order to detect weight-2 Z errors on the data qubits, which would result in a logical failure on the target $[\![15, 1, 3]\!]$ code. (b) Indexing of the bulk qubits used for the encoding circuit in alignment with the previous sections.

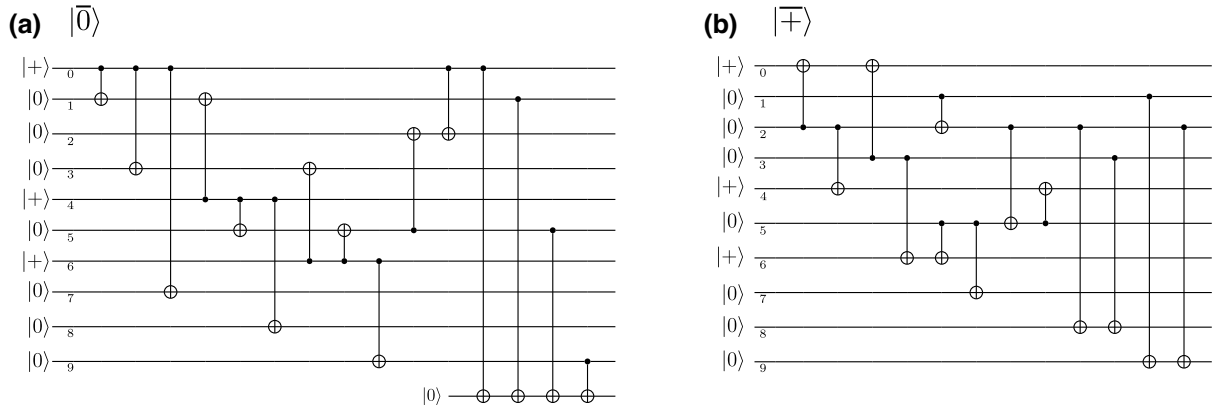

FIG. 23. Encoding circuit for the FT initialization of the morphed $[\![10,1,2]\!]$ code. Circuits for the FT initialization of (a) $|\overline{0}\rangle$ and (b) $|\overline{+}\rangle$ on the morphed $[\![10,1,2]\!]$ code. For the FT initialization of $|\overline{0}\rangle$, the logical operator $\overline{Z} = Z_0Z_1Z_5Z_9$ is measured to verify the resulting state. No verification step is required for the FT initialization of $|\overline{+}\rangle$, because every propagated Z error is detectable afterwards on the $[\![10,1,2]\!]$ code.

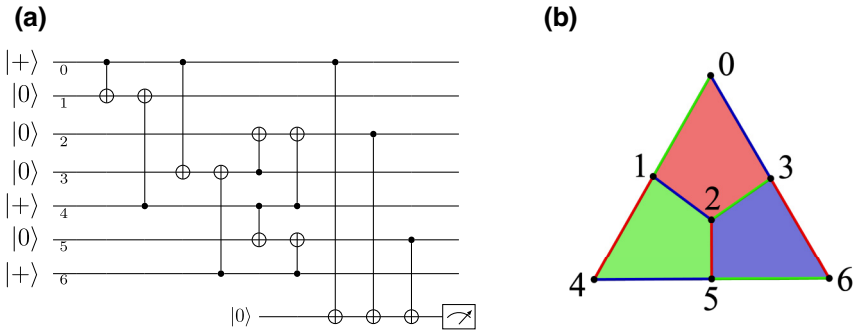

FIG. 24. Encoding circuit for the FT initialization of the $[\![7,1,3]\!]$ Steane code [39,40]. The logical operator $\overline{Z} = Z_0Z_2Z_5$ is measured in order to detect all single errors that propagate and would cause a logical failure.

[1] P. W. Shor, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (IEEE, Santa Fe, NM, USA, 1994), p. 124.

[2] L. K. Grover, Quantum computers can search arbitrarily large databases by a single query, Phys. Rev. Lett. **79**, 4709 (1997).

[3] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[4] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997, available online at https://thesis.library.caltech.edu/2900/2/THESIS.pdf.

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, Massachusetts, 2010).

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

[7] J. Preskill, Reliable quantum computers, Proc. Phys. Soc. A **454**, 385 (1998).

[8] A. Y. Kitaev, Quantum computations: Algorithms and error correction, Russ. Math. Surv. **52**, 1191 (1997).

[9] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2006).

[10] E. Knill, R. Laflamme, and W. H. Zurek, Resilient quantum computation, Science **279**, 342 (1998).

[11] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, SIAM J. Comput. **38**, 1207 (2008).

[12] N. D. Mermin, *Quantum Computer Science: An Introduction* (Cambridge University Press, Cambridge, England, UK, 2007).

[13] R. Solovay, Lie groups and quantum circuits (2000), MSRI presentation at http://www.msri.org/publica-tions/ln/msri/2000/qcomputing/solovay/1/. According to Ref. [43], the proof was announced by Solovay in 1995 on a virtual reading group e-mail discussion list.

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

[15] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt,

Quantum computations on a topologically encoded qubit, Science **345**, 302 (2014).
[16] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, Fault-tolerant quantum error detection, Sci. Adv. **3**, e1701074 (2017).
[17] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. Poschinger, Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer, Phys. Rev. X **12**, 011032 (2022).
[18] Y. Wan, R. Jördens, S. D. Erickson, J. J. Wu, R. Bowler, T. R. Tan, P.-Y. Hou, D. J. Wineland, A. C. Wilson, and D. Leibfried, Ion transport and reordering in a 2D trap array, Adv. Quantum Technol. **3**, 2000028 (2020).
[19] C. Ryan-Anderson, J. Bohnet, K. Lee, D. Gresh, A. Hankin, J. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, and N. Brown, *et al.*, Realization of real-time fault-tolerant quantum error correction, Phys. Rev. X **11**, 041058 (2021).
[20] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, Experimental demonstration of fault-tolerant state preparation with superconducting qubits, Phys. Rev. Lett. **119**, 180501 (2017).
[21] K. Satzinger, Y.-J. Liu, A. Smith, C. Knapp, M. Newman, C. Jones, Z. Chen, C. Quintana, X. Mi, and A. Dunsworth, *et al.*, Realizing topologically ordered states on a quantum processor, Science **374**, 1237 (2021).
[22] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, and J. Herrmann, *et al.*, Realizing repeated quantum error correction in a distance-three surface code, Nature **605**, 669 (2022).
[23] Google Quantum AI, Exponential suppression of bit or phase errors with cyclic error correction, Nature **595**, 383 (2021).
[24] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, H. Guan, Q. Zhu, Z. Wei, T. He, and S. Cao, *et al.*, Realization of an error-correcting surface code with superconducting qubits, Phys. Rev. Lett. **129**, 030501 (2022).
[25] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, Repeated quantum error detection in a surface code, Nat. Phys. **16**, 875 (2020).
[26] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, and Y. Chen, *et al.*, State preservation by repetitive error detection in a superconducting quantum circuit, Nature **519**, 66 (2015).
[27] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).
[28] A. Browaeys and T. Lahaye, Many-body physics with individually controlled Rydberg atoms, Nat. Phys. **16**, 132 (2020).
[29] M. Saffman, Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges, J. Phys. B **49**, 202001 (2016).
[30] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, and H. Pichler, *et al.*, Parallel implementation of high-fidelity multiqubit gates with neutral atoms, Phys. Rev. Lett. **123**, 170503 (2019).
[31] T. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, and B. Grinkemeyer, *et al.*, Multi-qubit entanglement and algorithms on a neutral-atom quantum computer, Nature **604**, 457 (2022).
[32] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, and N. Maskara, *et al.*, High-fidelity parallel entangling gates on a neutral atom quantum computer, ArXiv:2304.05420.
[33] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, and M. Greiner, *et al.*, A quantum processor based on coherent transport of entangled atom arrays, Nature **604**, 451 (2022).
[34] K. Takeda, A. Noiri, T. Nakajima, T. Kobayashi, and S. Tarucha, Quantum error correction with silicon spin qubits, Nature **608**, 682 (2022).
[35] M. Abobeih, Y. Wang, J. Randall, S. Loenen, C. Bradley, M. Markham, D. Twitchen, B. Terhal, and T. Taminiau, Fault-tolerant operation of a logical qubit in a diamond quantum processor, Nature **606**, 884 (2022).
[36] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, and M. Cetina, *et al.*, Fault-tolerant control of an error-corrected qubit, Nature **598**, 281 (2021).
[37] J. Marques, B. Varbanov, M. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, B. Alessandro, B. Terhal, and L. DiCarlo, Logical-qubit operations in an error-detecting surface code, Nat. Phys. **18**, 80 (2022).
[38] C. Ryan-Anderson, N. Brown, M. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. Bohnet, S. Braxton, and N. Burdick, *et al.*, Implementing fault-tolerant entangling gates on the five-qubit code and the color code, ArXiv:2208.01863.
[39] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, and R. Blatt, *et al.*, Demonstration of fault-tolerant universal quantum gate operations, Nature **605**, 675 (2022).
[40] H. Goto, Minimizing resource overheads for fault-tolerant preparation of encoded states of the Steane code, Sci. Rep. **6**, 1 (2016).
[41] C. Chamberland and A. W. Cross, Fault-tolerant magic state preparation with flag qubits, Quantum **3**, 143 (2019).
[42] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).
[43] A. Kubica and M. E. Beverland, Universal transversal gates with color codes: A simplified approach, Phys. Rev. A **91**, 032330 (2015).
[44] 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).
[45] H. Bombin, Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes, New J. Phys. **17**, 083002 (2015).
[46] H. Bombin and M. A. Martin-Delgado, Topological quantum distillation, Phys. Rev. Lett. **97**, 180501 (2006).
[47] 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).

[48] J. Dehaene and B. De Moor, Clifford group, stabilizer states, and linear and quadratic operations over GF (2), Phys. Rev. A **68**, 042318 (2003).
[49] A. Wu, K. Yin, A. W. Cross, A. Li, and Y. Ding, Enabling full-stack quantum computing with changeable error-corrected qubits, ArXiv:2305.07072.
[50] H. Bombin and M.-A. Martin-Delgado, Topological computation without braiding, Phys. Rev. Lett. **98**, 160502 (2007).
[51] R. Chao and B. W. Reichardt, Quantum error correction with only two extra qubits, Phys. Rev. Lett. **121**, 050502 (2018).
[52] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).
[53] T. Tansuwannont and D. Leung, Achieving fault tolerance on capped color codes with few ancillas, PRX Quantum **3**, 030322 (2022).
[54] R. S. Gupta, N. Sundaresan, T. Alexander, C. J. Wood, S. T. Merkel, M. B. Healy, M. Hillenbrand, T. Jochym-O'Connor, J. R. Wootton, and T. J. Yoder, *et al.*, Encoding a magic state with beyond break-even fidelity, ArXiv:2305.13581.
[55] M. Vasmer and A. Kubica, Morphing quantum codes, PRX Quantum **3**, 030319 (2022).
[56] D. Gottesman, Class of quantum error-correcting codes saturating the quantum Hamming bound, Phys. Rev. A **54**, 1862 (1996).
[57] 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).
[58] D. Gottesman, The Heisenberg representation of quantum computers, XXII International Colloquium on Group Theoretical Methods in Physics, 1999, https://arxiv.org/pdf/quant-ph/9807006.pdf.
[59] A. Steane, Multiple-particle interference and quantum error correction, Proc. Phys. Soc. A **452**, 2551 (1996).
[60] H. Bombin, Transversal gates and error propagation in 3D topological codes, ArXiv:1810.09575.
[61] A. M. Kubica, Ph.D. thesis, California Institute of Technology, 2018, available online at https://thesis.library.caltech.edu/10955/.
[62] S. Koutsioumpas, D. Banfield, and A. Kay, The smallest code with transversal T, ArXiv:2210.14066.
[63] D. Poulin, Stabilizer formalism for operator quantum error correction, Phys. Rev. Lett. **95**, 230504 (2005).
[64] D. Kribs, R. Laflamme, and D. Poulin, Unified and generalized approach to quantum error correction, Phys. Rev. Lett. **94**, 180501 (2005).
[65] 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).
[66] J. Preskill, Fault-tolerant quantum computers, 1998, https://authors.library.caltech.edu/3930/.
[67] Y. Aharonov, D. Z. Albert, and L. Vaidman, How the result of a measurement of a component of the spin of a spin-1/2 particle can turn out to be 100, Phys. Rev. Lett. **60**, 1351 (1988).
[68] A. Peres, Quantum measurements with postselection, Phys. Rev. Lett. **62**, 2326 (1989).
[69] E. T. Campbell, The smallest interesting colour code, Blog post, 2016, https://earltcampbell.com/2016/09/26/the-smallest-interesting-colour-code/.
[70] E. Barnes, C. Arenz, A. Pitchford, and S. E. Economou, Fast microwave-driven three-qubit gates for cavity-coupled superconducting qubits, Phys. Rev. B **96**, 024504 (2017).
[71] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. Poschinger, C. Hempel, and J. Home, *et al.*, Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation, Phys. Rev. X **7**, 041061 (2017).
[72] P. Parrado-Rodríguez, C. Ryan-Anderson, A. Bermudez, and M. Müller, Crosstalk suppression for fault-tolerant quantum error correction with trapped ions, Quantum **5**, 487 (2021).
[73] C. Ryan-Anderson, PECOS: Performance estimator of codes on surfaces, 2019, https://github.com/PECOS-packages/PECOS.
[74] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S.-T. Wang, L. Duan, and K. R. Brown, Simulating the performance of a distance-3 surface code in a linear ion trap, New J. Phys. **20**, 043038 (2018).
[75] S. Heußen, L. Postler, M. Rispler, I. Pogorelov, C. D. Marciniak, T. Monz, P. Schindler, and M. Müller, Strategies for a practical advantage of fault-tolerant circuit design in noisy trapped-ion quantum computers, Phys. Rev. A **107**, 042422 (2023).
[76] H. Bombín, Dimensional jump in quantum error correction, New J. Phys. **18**, 043038 (2016).
[77] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger, Shuttling-based trapped-ion quantum information processing, AVS Quantum **2**, 014101 (2020).
[78] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. Allman, C. Baldwin, M. Foss-Feig, D. Hayes, and K. Mayer, *et al.*, Demonstration of the trapped-ion quantum CCD computer architecture, Nature **592**, 209 (2021).
[79] S. Bravyi, O. Dial, J. M. Gambetta, D. Gil, and Z. Nazario, The future of quantum computing with superconducting qubits, J. Appl. Phys. **132**, 160902 (2022).
[80] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum **4**, 327 (2020).
[81] S. Moses, C. Baldwin, M. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, and M. Bohn, *et al.*, A race track trapped-ion quantum processor, ArXiv:2305.03828.
[82] A. K. Pal, P. Schindler, A. Erhard, Á. Rivas, M.-A. Martin-Delgado, R. Blatt, T. Monz, and M. Müller, Relaxation times do not capture logical qubit dynamics, Quantum **6**, 632 (2022).
[83] E. Huang, A. Pesah, C. T. Chubb, M. Vasmer, and A. Dua, Tailoring three-dimensional topological codes for biased noise, ArXiv:2211.02116.
[84] A. Jain, P. Iyer, S. Bartlett, and J. Emerson, Improved quantum error correction with randomized compiling, Bull. Am. Phys. Soc. **5**, 033049 (2023).
[85] A. M. Steane, Fast fault-tolerant filtering of quantum codewords, ArXiv:quant-ph/0202036.