npj | Quantum Information

ARTICLE OPEN

Check for updates

Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits

Christopher Chamberland[1,2] ✉ and Kyungjoo Noh[1] ✉

Fault-tolerant quantum computing promises significant computational speedup over classical computing for a variety of important problems. One of the biggest challenges for realizing fault-tolerant quantum computing is preparing magic states with sufficiently low error rates. Magic state distillation is one of the most efficient schemes for preparing high-quality magic states. However, since magic state distillation circuits are not fault-tolerant, all the operations in the distillation circuits must be encoded in a large distance error-correcting code, resulting in a significant resource overhead. Here, we propose a fault-tolerant scheme for directly preparing high-quality magic states, which makes magic state distillation unnecessary. In particular, we introduce a concept that we call redundant ancilla encoding. The latter combined with flag qubits allows for circuits to both measure stabilizer generators of some code, while also being able to measure global operators to fault-tolerantly prepare magic states, all using nearest neighbor interactions. We apply such schemes to a planar architecture of the triangular color code family and demonstrate that our scheme requires at least an order of magnitude fewer qubits and space–time overhead compared to the most competitive magic state distillation schemes. Since our scheme requires only nearest-neighbor interactions in a planar architecture, it is suitable for various quantum computing platforms currently under development.

npj Quantum Information (2020)6:91 ; https://doi.org/10.1038/s41534-020-00319-5

INTRODUCTION

Ideal quantum computers can efficiently solve several important problems of great interest. Such problems include (but are not limited to) integer factoring[1] and simulation of large quantum systems[2] and are believed to be not solvable in polynomial time in the problem size by using classical computers. However, due to inevitable noise in realistic quantum systems, currently available quantum devices are not capable of performing long quantum computations in a reliable way. Quantum error correction and fault-tolerant techniques are thus essential to build a universal quantum computer (capable of implementing arbitrary quantum algorithms) out of noisy physical components. Importantly, due to significant hardware constraints, the resource overhead associated with the implementation of fault-tolerant methods will need to be low.

The transversal implementation of a logical gate is naturally fault-tolerant since errors cannot spread between qubits in a given code-block. Unfortunately, from the Eastin–Knill theorem[3], given any quantum error-correcting code, there will always be at least one gate in a universal gate set that cannot be implemented using transversal operations at the error-corrected logical level. Several fault-tolerant methods for implementing gates in a universal gate set have been proposed[4–17]. Among these various proposals, preparing high-quality magic states by using magic state distillation has remained a leading candidate for realizing universal fault-tolerant quantum computing[18–27].

It had long been believed that implementing magic state distillation was the dominant cost of building a universal fault-tolerant quantum computer. While recent results have shown that this is not necessarily the case[28], the cost of performing magic state distillation still remains very high given current hardware constraints. One of the reasons for the high costs of magic state distillation is that the distillation circuits are generally not robust against failures of the Clifford gates used to distill magic states. Consequently, the Clifford gates must be encoded in some large-distance error-correcting code (often the surface code) to ensure that these gates have negligible error rates compared to the magic states being injected. Due to the high threshold of the surface code, magic state distillation schemes with the surface code can be implemented with noise rate near $p = 10^{-3}$ using a large enough code distance.

In ref. [15], a fault-tolerant method for directly preparing an H-type magic state was proposed using the Steane code and flag-qubit circuits[29–38]. For physical error rates $p \gtrsim 10^{-5}$ and with idle qubits failing with error rates 100 times smaller than single-qubit gate error rates, it was shown that fewer qubits were required to prepare H-type magic states than the best-known distillation schemes. Unfortunately, the scheme requires the ability to perform geometrically non-local gates and is scaled up by concatenating the Steane code with itself, making it difficult to implement such a scheme in a scalable way with realistic quantum hardware.

The Steane code belongs to the family of two-dimensional color codes[8,39–41], which are topological codes. In particular, the two-dimensional color codes have a nice property that all logical Clifford gates can be implemented using transversal operations. In particular, for a color code with n data qubits, the logical Hadamard gate is simply given by $\overline{H} = H^{\otimes n}$. This features makes color codes particularly well suited for preparing H-type magic states. Furthermore, recent work introduced a simple and efficient decoding algorithm for color codes which also uses information from flag qubits to decode up to the full effective code distance[38,42].

In this work, we introduce a fault-tolerant scheme for directly preparing an H-type magic state encoded in the triangular color code family. In particular, we introduce a concept that we call

[1]AWS Center for Quantum Computing, Pasadena, CA 91125, USA. [2]Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, CA 91125, USA.
✉email: cchmber@amazon.com; nkyungjo@amazon.com

Published in partnership with The University of New South Wales

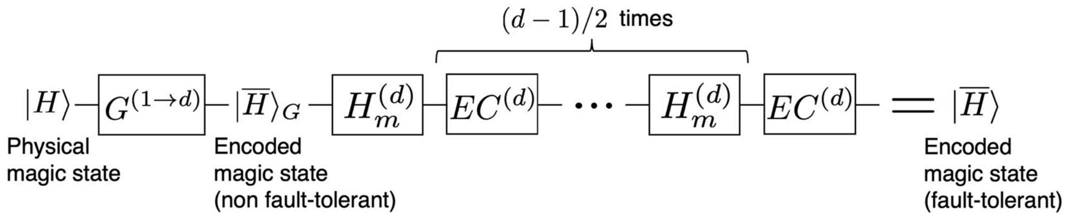

Fig. 1 Overview of our fault-tolerant magic state preparation scheme. To prepare an H-type magic state encoded in the distance-d triangular color code (i.e., $|\overline{H}\rangle$), we first grow a physical $|H\rangle$ state non-fault-tolerantly to an encoded state which we label $|\overline{H}\rangle_G$. We represent the circuit that performs the non-fault-tolerant growing operation by the label $G^{(1\to d)}$. An example for the $G^{(1\to d)}$ circuit is given in Fig. 7. Second, a t-flag circuit (with $t=(d-1)/2$) for performing a non-destructive measurement of $\overline{H}=H^{\otimes n}$ (labeled as $H_m^{(d)}$) is applied. In particular, a GHZ ancilla is constructed in a fault-tolerant way to measure the global operator $H^{\otimes n}$. An example for one round of the application of the $H_m^{(d)}$ circuit is provided in Figs. 3 and 4. Third, error correction for the distance d triangular color code (circuit labeled $EC^{(d)}$) is applied immediately following the $H_m^{(d)}$ circuit (see Fig. 2 for an example when $d=5$). If any flag qubits flag, or the parity of the ancilla measurements in either $H_m^{(d)}$ or $EC^{(d)}$ is odd, the protocol is aborted. To guarantee fault-tolerance, each pair of $H_m^{(d)}$ and $EC^{(d)}$ circuits need to be repeated $(d-1)/2$ times (see Supplementary Note 2 and Supplementary Fig. 2 for the proof of the fault-tolerance).

redundant ancilla encoding. The latter combined with flag qubits allows for circuits to fault-tolerantly extract error syndrome information and measure the global logical Hadamard operator to fault-tolerantly prepare magic states. Due to the fault-tolerance of our proposed scheme, magic states with very low logical error rates can be directly prepared without the need to use very large-distance error-correcting codes to encode the required Clifford operations. That is, we find that the Clifford gates in our scheme can have error rates comparable to the magic states being injected, thus significantly reducing the resource requirements. For instance, for a physical error rate $p=10^{-4}$, to produce a magic state with a logical error rate of ~5×10^{-8}, our scheme requires only 64 qubits (compared to 810 with the best known magic state distillation schemes[28]) and all Clifford gate operations can be performed at the physical level. Thus, our scheme requires more than an order of magnitude fewer qubits and space–time overhead costs compared to the most competitive existing magic state distillation schemes. Most importantly, our scheme uses only nearest-neighbor interactions in a two-dimensional planar architecture and thus is well suited for various quantum computing platforms that are currently under development. Moreover, our scheme can be scaled to produce magic states with arbitrarily low error rates by using logical Clifford operations encoded in a two-dimensional topological code. We find that using the surface code for our encoded stabilizer operations appears to be the most resource-efficient scheme for producing magic states with error rates lower than 10^{-15}.

RESULTS

Overview of our proposed magic state preparation scheme

Note that logical Clifford gates encoded in a color code can be implemented using transversal operations and hence are fault-tolerant. Thus, to complete a universal gate set, we focus on fault-tolerantly preparing an H-type magic state encoded in a triangular color code, which can be used to implement the non-Clifford T gate (see the Methods section for more details on the Clifford and non-Clifford gates, stabilizer operations, H-type magic states, and the triangular color code). Depending on the size and duration of a quantum algorithm, the desired probability that an $|H\rangle$ state is afflicted by an error ranges from 10^{-7} to ~10^{-15} (see, e.g., refs. [43–45]). Since an H-type magic state $|H\rangle$ is an eigenstate of the Hadamard operator H with $+1$ eigenvalue, a high-quality H-type magic state can be prepared if the Hadamard operator H can be measured non-destructively and in a fault-tolerant way. Here, we propose a fault-tolerant scheme for measuring the logical Hadamard operator $\overline{H}=H^{\otimes n}$ of the triangular color code, which can be used to directly prepare a high-quality magic state.

An overview of our fault-tolerant magic state preparation scheme is provided in Fig. 1. Our scheme outputs an H-type magic state $|\overline{H}\rangle$ encoded in the distance-d triangular color code and consists of the following three elements. $G^{(1\to d)}$: a non-fault-tolerant growing operation of a physical $|H\rangle$ state to an encoded state $|\overline{H}\rangle_G$ (see Fig. 7 and the Methods section), $H_m^{(d)}$: a fault-tolerant logical Hadamard measurement circuit (for $d\in\{3,5,7\}$; see Figs. 3 and 4), and $EC^{(d)}$: a fault-tolerant syndrome extraction circuit (see Fig. 2). In particular, we call the latter two circuits $H_m^{(d)}$ and $EC^{(d)}$ fault-tolerant in the sense that they are t-flag circuits with $t=(d-1)/2$, where the definition of t-flag circuit is given below:

Definition 2 (t-flag circuit) *A circuit C(U) which, when fault-free, implements a projective measurement of a weight-w operator $U\in\mathcal{P}_n^{(2)}$ without flagging is a t-flag circuit if the following holds: For any set of v faults at up to t locations in C(U) resulting in an error E with* $\min(wt(E),wt(EU))$, *the circuit flags (i.e., at least one flag qubit has a non-trivial measurement outcome).*

In our scheme, we first grow a noisy physical $|H\rangle$ state to an encoded state $|\overline{H}\rangle_G$ via a growing operation $G^{(1\to d)}$. Due to the non-fault-tolerance of the growing operation, a single fault can result in a logical error afflicting $|\overline{H}\rangle_G$. However, these errors can be detected by implementing in pairs a fault-tolerant logical Hadamard measurement circuit $H_m^{(d)}$ and a syndrome extraction circuit $EC^{(d)}$ $(d-1)/2$ times. Specifically, the Hadamard measurement circuit $H_m^{(d)}$ checks for logical errors in the encoded magic state $|\overline{H}\rangle_G$ and the syndrome extraction circuit $EC^{(d)}$ prevents the accumulation of residual errors so that they remain correctable by the code[15,46]. In particular, we show in Supplementary Note 2 that our magic state preparation scheme in Fig. 1 is fault-tolerant in the following sense[15,47]:

Definition 2 (Fault-tolerant state preparation) *For $t=(d-1)/2$, a state-preparation protocol using a distance-d stabilizer code C is said to be fault-tolerant if the following two conditions are satisfied: (1) If there are s faults during the state-preparation protocol with $s\le t$, the resulting state differs from a codeword by an error of at most weight s. (2) If there are s faults during the state-preparation protocol with $s\le t$, then ideally decoding the output state results in the same state that would be obtained from the fault-free state-preparation scheme.*

Thus, $(d-1)/2$ faults in a distance $d\in\{3,5,7\}$ triangular color code of our magic state preparation scheme cannot result in a logical failure. In what follows, we describe the syndrome extraction circuit $EC^{(d)}$ and the Hadamard measurement circuit $H_m^{(d)}$ in more detail.

Syndrome extraction with redundant ancilla encoding

Here, we describe the error syndrome extraction circuit $EC^{(d)}$ used to detect errors at the output of the Hadamard measurement

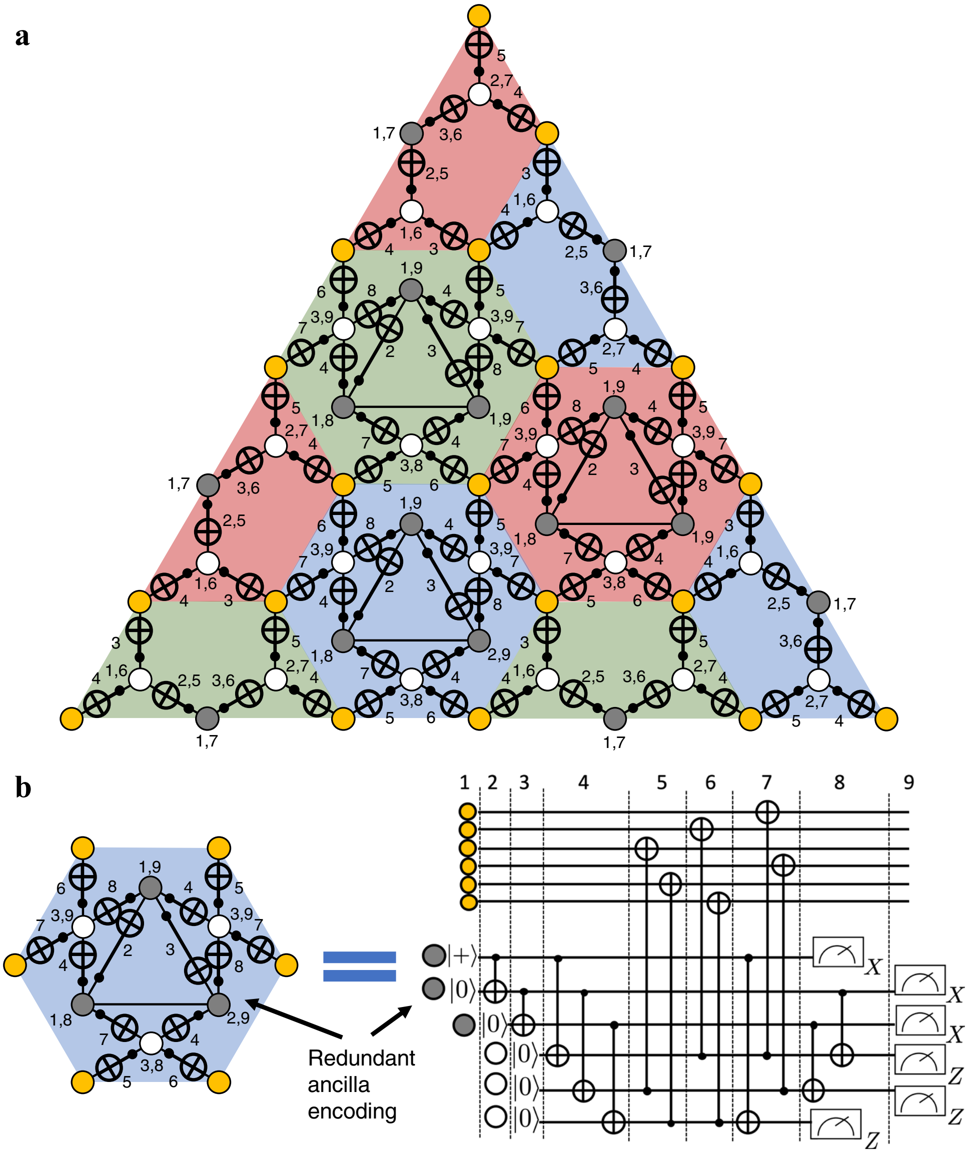

Fig. 2 Fault-tolerant error syndrome extraction circuit. a $EC^{(5)}$ circuit for measuring the X stabilizers of the $d = 5$ color code. The gray circles correspond to the ancilla qubits used to measure the parity of the stabilizers, whereas the white circles correspond to the flag qubits. The Z stabilizers can be measured similarly by reversing the direction of the CNOT gates and replacing $|0\rangle$ ($|+\rangle$) states by $|+\rangle$ ($|0\rangle$) states, and Pauli X (Z) measurements by Pauli Z (X) measurements. The CNOT gate scheduling that minimizes the total number of time steps for measuring the X and Z-stabilizers is also provided. One round of X or Z-type stabilizer measurements requires a total of nine time steps. **b** Circuit for measuring a weight-six X-type stabilizer of the triangular color code. Note that a redundant amount of ancillas are used since they become flag qubits when measuring the logical Hadamard operator $H^{\otimes n}$ of the triangular color code (see Fig. 4).

circuit $H_m^{(d)}$. In particular, to guarantee fault-tolerance of our scheme (see Definition 2), we require that the $EC^{(d)}$ circuit is at least a one-flag circuit. For instance, if the circuit is not a one-flag circuit, two single faults in two separate stabilizers can result in a weight-four data qubit error without any flag qubits flagging. Such an error is thus not rejected by the circuit and might not be correctable by the distance-5 triangular color code, despite being derived from a two-fault event.

In Fig. 2a we illustrate the full $EC^{(5)}$ circuit used to measure the stabilizers of a triangular color code along with the CNOT gate scheduling, which minimizes the circuit depth for the given qubit layout. The weight-four stabilizers are identical to the ones used in ref. [38]. However, an important difference can be observed for the weight-six stabilizers, which in addition to using three flag qubits, also uses three ancilla qubits. If an error anti-commutes with a weight-six stabilizer, the measurement outcomes of the three ancillas will have odd parity, otherwise it will have even parity. The circuit for measuring a weight-six X-type stabilizer that respects the CNOT scheduling is given in Fig. 2b. By performing an exhaustive numerical search, we verified that both weight-four and weight-six circuits are two-flag circuits. Since the weights of the stabilizer generators are independent of d, having two-flag circuits is sufficient to ensure that the $EC^{(d)}$ circuits are implemented fault-tolerantly[32]. Lastly, each weight-four and weight-six plaquette in a general $EC^{(d)}$ circuit has the same qubit layout and gate connectivity as those of Fig. 2a.

Now, one might wonder why three ancilla qubits (gray circles in Fig. 2b) instead of a single ancilla are used for measuring the weight-six stabilizers (see for instance the weight-six circuits used in ref. [38] that only require one ancilla). Indeed, if one is only interested in performing fault-tolerant error correction instead of fault-tolerant quantum computation, then a single ancilla qubit

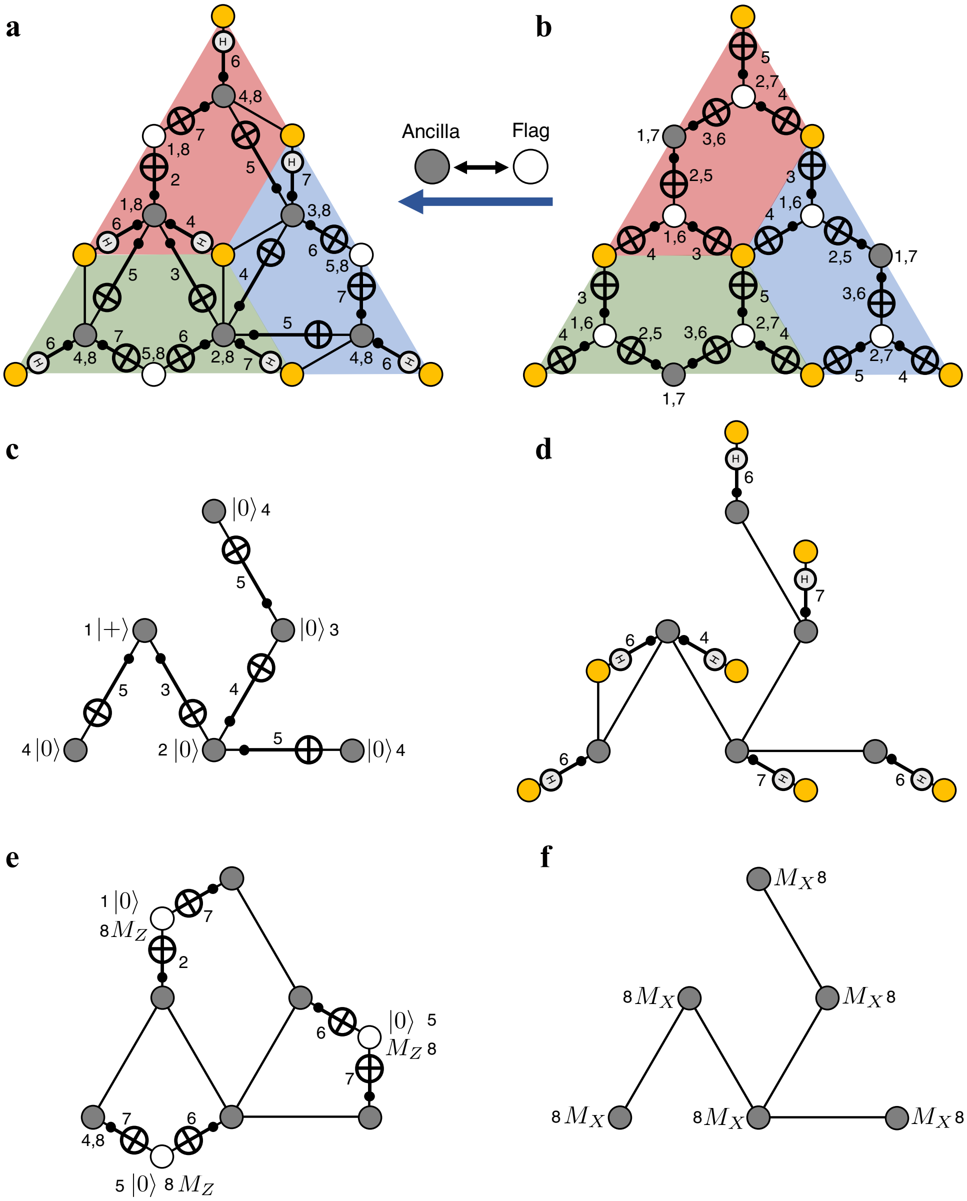

Fig. 3 Fault-tolerant logical Hadamard measurement for the distance-3 triangular color code. a A one-flag circuit $H_m^{(3)}$ for measuring the logical Hadamard operator $H^{\otimes 7}$ of the $d = 3$ triangular color code. **b** A circuit for measuring the X-type stabilizers of the $d = 3$ triangular color code. Note that there is a role reversal between the ancilla qubits (gray circles) and the flag qubits (white circles). Also, the controlled-H symbols in **a** represent the controlled-Hadamard gates. The logical Hadamard circuit $H_m^{(3)}$ in **a** can be decomposed into four elements: **c** Preparation of the 6-qubit GHZ state $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes 6} + |1\rangle^{\otimes 6})$ (or the logical plus state of the 6-qubit repetition code) in the ancilla qubits (gray circles). **d** Controlled-Hadamard gates between the ancilla qubits (in the GHZ state) and the data qubits (yellow circles). **e** Parity checks of the ancilla GHZ state by using flag qubits (white circles). **f** Measurement of the logical X operator $X^{\otimes 6}$ of the 6-qubit repetition code on the ancilla qubits.

suffices since the additional ancillas do not provide more information and they also increase the circuit depth. However, we show below that to use the same qubit layout for measuring the logical Hadamard operator $\overline{H} = H^{\otimes n}$ (which is a global operator), the roles of the ancillas and flags in Fig. 2a are reversed. In other words, the ancilla qubits become flag qubits and the flag qubits become ancilla qubits. We thus use more ancilla qubits than necessary to ensure the Hadamard measurement circuit $H_m^{(d)}$ satisfies the desired fault-tolerant properties. We refer to this extra redundancy as redundant ancilla encoding. Below, we demonstrate the utility of using redundant ancilla encoding.

Fault-tolerant logical Hadamard measurement

Here, we describe in detail how to construct a t-flag circuit (with $t = (d - 1)/2$) for the non-destructive measurement of the logical Hadamard operator (i.e., $H_m^{(d)}$). The circuits we construct apply to distance d triangular color codes with $d \in \{3, 5, 7\}$. In Fig. 3a, we present a one-flag circuit for measuring of the logical Hadamard operator $H^{\otimes 7}$ of the $d = 3$ triangular color code. Note that we used the same qubit layout as the one used for the stabilizer measurement circuit ($EC^{(3)}$ circuit shown in Fig. 3b) but there is a role reversal between the ancilla qubits (gray circles) and the flag qubits (white circles).

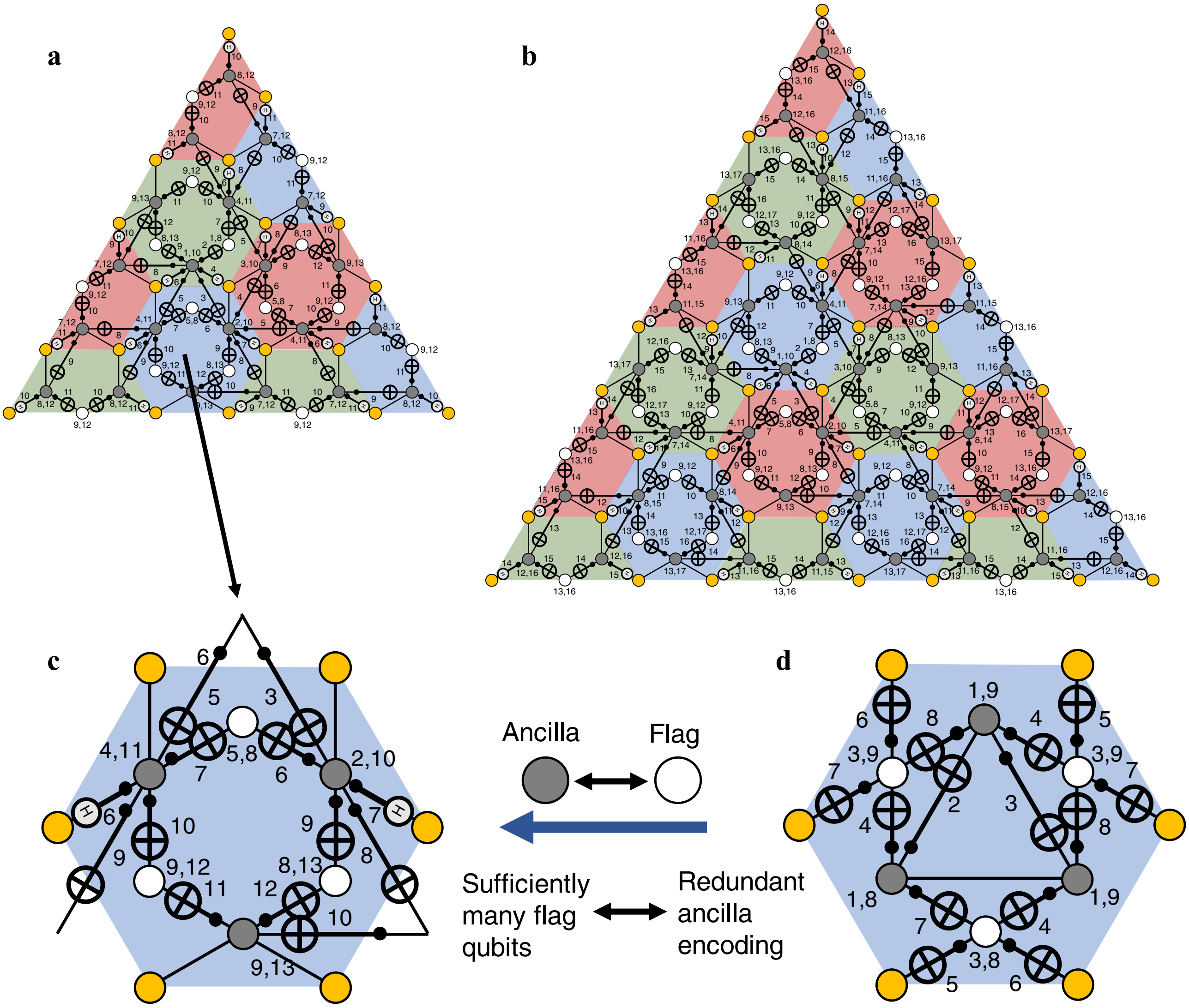

Fig. 4 Fault-tolerant logical Hadamard measurement for the distance-5 and 7 triangular color codes. **a** A two-flag circuit $H_m^{(5)}$ for measuring the logical Hadamard operator $H^{\otimes 19}$ of the $d = 5$ triangular color code and **b** a three-flag circuit $H_m^{(7)}$ for measuring the logical Hadamard operator $H^{\otimes 37}$ of the $d = 7$ triangular color code. **c** Illustration of the benefit of using redundant ancilla encoding. Note that the ancilla qubits in the stabilizer measurement circuits are used as flag qubits in the logical Hadamard measurement circuits. Thus, redundant ancilla encoding in the stabilizer measurement circuits (shown in **d**) allows the logical Hadamard measurement circuits to have sufficiently many flag qubits.

We now explain why the circuit in Fig. 3a performs a non-destructive measurement of the logical Hadamard operator $H^{\otimes 7}$. Note that the ancilla qubits (gray circles) in Fig. 3a are prepared in the logical plus state of the 6-qubit repetition code (or the 6-qubit GHZ state $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes 6} + |1\rangle^{\otimes 6})$) through the CNOT gates between the ancilla qubits (see Fig. 3c). Then, as shown in Fig. 3d, the 7 controlled-Hadamard gates implement a logical controlled-Hadamard gate between the ancilla 6-qubit repetition code and the data $d = 3$ triangular color code. Eventually, the ancilla qubits (in the 6-qubit repetition code) are measured in the logical X basis via a $X^{\otimes 6}$ measurement (see Fig. 3f). Hence, the circuits in Fig. 3c, d, and f implement the simple non-destructive Hadamard measurement circuit in Fig. 5b, except that the ancilla qubits are now encoded in the 6-qubit repetition code and the data qubits are encoded in the $d = 3$ triangular color code.

The most important element of the logical Hadamard measurement circuit in Fig. 3 is the parity check of the ancilla GHZ state by using flag qubits (see Fig. 3e). Note that the flag qubits (white circles) non-destructively measure the parity of the ancilla GHZ state Z_1Z_2, Z_3Z_6, and Z_4Z_5, or the three (out of five) stabilizers of the 6-qubit repetition code (we labeled the qubits from the top to the bottom and from the left to the right). These stabilizer measurements will be trivial if all the CNOT gates in Fig. 3c are perfect. However, CNOT gate failures can result in non-trivial flag measurement outcomes. In particular, there are several single CNOT gate failure events that can cause a data qubit error of weight-two or higher. For the Hadamard circuit to be one-flag, all these failure events should be caught by flag qubits. Indeed, we verify via a comprehensive numerical search that the logical Hadamard circuit in Fig. 3a is a one-flag circuit by confirming that if a single fault at any location results in a data qubit error E with $\min(wt(E), wt(EH^{\otimes 7}))>1$, at least one flag qubit flags.

Further, we construct a two-flag circuit for the logical Hadamard measurement $H_m^{(5)}$ of the $d = 5$ triangular color code and a three-flag circuit $H_m^{(7)}$ for the $d = 7$ triangular color code (see Fig. 4). Designing such circuits with the desired fault-tolerant properties is very involved as it requires careful utilization of the ample flag qubits that become available due to redundant ancilla encoding. However, at a high level, the design principle is similar to the one used for the $d=3$ case. That is, in the $d=5$ case (i.e., $H_m^{(5)}$), we first generate a 21-qubit GHZ state in the ancilla qubits (gray circles), or equivalently, a logical plus state of the 21-qubit repetition code. The entangled ancilla qubits are then coupled to all of the 19 data qubits (yellow circles) through controlled-Hadamard gates. Finally, the ancilla qubits (in the 21-qubit repetition code) are measured in the logical X basis via a $X^{\otimes 21}$ measurement. Thus, the circuit in Fig. 4a implements a non-destructive measurement of the logical Hadamard measurement of the $d=5$ color code by using ancilla qubits in the 21-qubit repetition code.

Because the ancilla qubits are encoded in the 21-qubit repetition code, we can check if the ancilla qubits are reliably prepared in the logical plus state by measuring 15 (out of 20) stabilizers of the ancilla 21-qubit repetition code using 15 flag qubits (white circles). Performing an exhaustive numerical search, we can confirm that the circuit $H_m^{(5)}$ is a two-flag circuit. A three-flag circuit for the logical Hadamard measurement $H_m^{(7)}$ of the $d=7$ triangular color code is constructed in the same way by using 45 ancilla qubits prepared in the logical plus state of the 45-qubit repetition code, and using 36 flag qubits checking the 36 (out of 44) stabilizers of the 45-qubit repetition code (see Fig. 4b). We point out that the order and time steps at which the CNOT gates (used to measure the stabilizers of the repetition code or the GHZ state with flag qubits) are implemented is very important and must be carefully chosen.

Recall that we used a redundant ancilla encoding scheme in the stabilizer measurement circuits (see Fig. 2) and the ancilla qubits in the stabilizer measurement circuits are used as flag qubits in the Hadamard measurement circuits. Thus, redundant ancilla encoding in the stabilizer measurement circuits allows the logical Hadamard measurement circuits to have sufficiently many flag qubits while maintaining the same two-dimensional qubit layout. In particular, the logical Hadamard measurement circuits $H_m^{(5)}$ and $H_m^{(7)}$ have 15 and 36 flag qubits that check 15 (out of 20) and 36 (out of 45) stabilizers of the ancilla GHZ state, respectively. Without the redundant ancilla encoding, we would have had 9 flag qubits in the $d=5$ case and 18 flag qubits in the $d=7$ case, as opposed to 15 and 36 flag qubits, respectively. We remark that the logical Hadamard measurement circuits $H_m^{\prime(5)}$ and $H_m^{\prime(7)}$ that are constructed with such fewer flag qubits are not two-flag and three-flag circuits. Thus, the redundant ancilla encoding scheme plays a crucial role in guaranteeing the desired fault-tolerance property of the logical Hadamard measurement circuits. We also remark that in previous works[46,48], it has been shown that GHZ ancilla states can be used to fault-tolerantly measure the logical Hadamard operator. However, such GHZ states need to be prepared using ancilla verification methods, which can be very costly and typically requires very low noise rates and non-local interactions.

Performance of the scheme with physical operations

Since our magic state preparation scheme in Fig. 1 is fault-tolerant, it is robust against any v faults with $v \le (d-1)/2$ (including failures of all stabilizer operations). Consequently, under the circuit-level noise model detailed in the Methods section, the logical error probability of the output magic state $|\overline{H}\rangle$ is given by

$$p_L^{(d)} = ap^{(d+1)/2} + \mathcal{O}(p^{(d+3)/2}), \qquad (1)$$

where p is the physical error rate and a counts all combinations of $(d+1)/2$ faults that lead to acceptance in our state-preparation scheme but results in a logical $\overline{X}$, $\overline{Y}$, or $\overline{Z}$ error. We numerically find that for values of $p \le 10^{-3}$, the higher-order terms in Eq. (1) were found to have a negligible impact on p_L (see Table 1).

Note that the logical error rate in Eq. (1) is conditioned on acceptance of the output magic state produced by our state-preparation scheme, i.e., when all the flag qubits do not flag and all the ancilla qubits have even parity. In the "Methods" section,

Table 1. Performance of our magic state preparation scheme using physical stabilizer operations.

Code distance for $\lvert\overline{H}\rangle$	Physical error rate p	Logical error rate $p_L^{(d)}$	Average qubit overhead $\langle n_{tot}^{(d)}\rangle$	Minimum qubit overhead $\min(n_{tot}^{(d)})$	Space–time overhead $s_O^{(d)}(p)$
$d=3$	10^{-4}	3.45×10^{-6}	17	16	594
$d=5$	10^{-4}	3.6×10^{-8}	68	55	5694
$d=7$	10^{-4}	$^*4.9\times10^{-10}$	231	118	27,431
$d=3$	2×10^{-4}	1.39×10^{-5}	17	16	611
$d=5$	2×10^{-4}	3.01×10^{-7}	84	55	7010
$d=7$	2×10^{-4}	$^*7.83\times10^{-9}$	449	118	53,359
$d=3$	3×10^{-4}	3.11×10^{-5}	18	16	630
$d=5$	3×10^{-4}	1.10×10^{-6}	103	55	8648
$d=7$	3×10^{-4}	$^*3.97\times10^{-8}$	870	118	103,500
$d=3$	4×10^{-4}	5.64×10^{-5}	18	16	650
$d=5$	4×10^{-4}	2.48×10^{-6}	127	55	10,656
$d=7$	4×10^{-4}	$^*1.25\times10^{-7}$	1700	118	202,268
$d=3$	5×10^{-4}	8.51×10^{-5}	19	16	670
$d=5$	5×10^{-4}	5.23×10^{-6}	156	55	13,115
$d=7$	5×10^{-4}	$^*3.06\times10^{-7}$	3312	118	394,177

Logical error rate p_L, average number of qubits $\langle n_{tot}^{(d)}\rangle$ (see Eqs. (1) and (4)), minimum number of qubits (Eq. (5)), and the space–time overhead (Eq. (6)) of the $|\overline{H}\rangle$ state preparation scheme in Fig. 1 obtained from 10^9 Monte–Carlo simulations using the circuit-level noise model in the "Methods" section.

*For $d=7$, we obtained five data points in the interval $p \in [3\times10^{-4}, 4\times10^{-4}]$ and extrapolated the best fit curve using Eq. (1) to obtain all the data in this table.

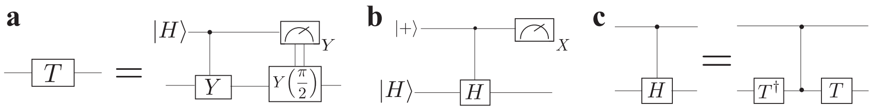

Fig. 5 Basic properties of H-type magic states. **a** Circuit for simulating a T gate using one copy of an $|H\rangle$ state and stabilizer operations. If the Y-basis measurement outcome is -1, a $Y(\frac{\pi}{2})$ gate is applied to the data qubit, otherwise $Y(\frac{\pi}{2})$ is not applied. **b** Circuit used to implement a non-destructive measurement of the Hadamard operator. **c** Decomposition of the controlled Hadamard gate in terms of non-Clifford T and $T^{\dagger}$ gates and a controlled-Z gate (which belongs to the Clifford group).

we described in detail how to compute the acceptance probability $p_{acc}^{(d)}(p)$. Here, we focus on estimating the qubit overhead and the space–time overhead of our magic state distillation scheme.

Given the acceptance probability $p_{acc}^{(d)}(p)$ for preparing a distance d encoded state $|\overline{H}\rangle$ with physical error rate p, the average number of qubits is given by

$$\langle n_{tot}^{(d)}(p)\rangle = \frac{n^{(d)} + n_{anc}^{(d)}}{p_{acc}^{(d)}(p)}, \tag{2}$$

where $n^{(d)} = (3d^2+1)/4$ is the number of data qubits, and $n_{anc}^{(d)}$ is the total number of ancilla and flag qubits used in the circuits $H_m^{(d)}$ and $EC^{(d)}$. Since each weight-six stabilizer generator requires six qubits, and each weight-four stabilizer three qubits, we have

$$n_{anc}^{(d)} = 6n_{w_6}^{(d)} + 3n_{w_4}^{(d)}, \tag{3}$$

where $n_{w_4}^{(d)} = (3/2)(d-1)$ and $n_{w_6}^{(d)} = (3d^2-12d+9)/8$ are the number of weight-four and weight-six stabilizers. Putting everything together, we obtain

$$\langle n_{tot}^{(d)}(p)\rangle = \frac{6d^2-9d+5}{2p_{acc}^{(d)}(p)}. \tag{4}$$

Note that if we repeat the protocol for preparing $|\overline{H}\rangle$ until the output state is accepted (at the expense of having a higher time cost), the number of qubits required to prepare $|\overline{H}\rangle$ is simply

$$\min(n_{tot}^{(d)}) = \frac{6d^2-9d+5}{2}. \tag{5}$$

We now consider the space–time overhead for implementing the scheme in Fig. 1, where additional qubits (as in Eq. (4)) are used to minimize the time cost. In particular, the space–time overhead $s_O^{(d)}(p)$ is given as

$$s_O^{(d)}(p) = \langle n_{tot}^{(d)}(p)\rangle(14 + \frac{(d-1)}{2}(t_{H_m}^{(d)} + t_{EC}^{(d)})), \tag{6}$$

where $t_{H_m}^{(d)}$ and $t_{EC}^{(d)}$ are the total number of time steps required to implement one round of the circuits $H_m^{(d)}$ and $EC^{(d)}$ respectively. The factor of 14 in Eq. (6) comes from the 14 times steps required to implement the circuit $|\overline{H}\rangle_G$. Note that we pessimistically assume that all time steps are reached when implementing the circuits in Fig. 1. This is pessimistic since when the magic state preparation scheme does not pass the verification steps, fewer than $14 + \frac{(d-1)}{2}(t_{H_m} + t_{EC})$ time steps are used.

The values of $p_L^{(d)}$, $\langle n_{tot}^{(d)}(p)\rangle$, $\min(n_{tot}^{(d)})$ and $S_O^{(d)}(p)$ for $p \in [10^{-4}, 5\times10^{-4}]$ are given in Table 1 and were obtained by performing 10^9 Monte–Carlo simulations on AWS clusters. For values of p resulting in very low logical failure rates, extrapolation of the best fit curve (using Eq. (1)) was used to compute the logical error rate of $|\overline{H}\rangle$. Note that using the $d=5$ version of the protocol in Fig. 1, an $|\overline{H}\rangle$ state can be prepared with a logical failure rate of 3.6×10^{-8} while using only 68 qubits on average when $p=10^{-4}$. Alternatively, one may use $\min(n_{tot}^{(d=5)}) = 55$ qubits and repeat the protocol until it is accepted (with an acceptance probability $p_{acc}^{(d=5)}(p=10^{-4}) = 0.81$). Further, one can use the $d=7$ version of the protocol to produce an $|\overline{H}\rangle$ state with logical failure rate 4.9×10^{-10} with only 231 qubits on average. Similarly as above, one may use $\min(n_{tot}^{(d=7)}) = 118$ qubits and repeat the protocol until it is accepted (with an acceptance probability $p_{acc}^{(d=7)}(p=10^{-4}) = 0.51$). In comparison, when $p=10^{-4}$, the non fault-tolerant magic state distillation scheme in ref. [28] requires 810 and 1150 qubits to prepare magic states with failure rates 4.4×10^{-8} and 9.3×10^{-10}, respectively. We note that recent work suggests that $p=10^{-4}$ is the appropriate regime to have small enough decoding hardware requirements[49].

An important remark is that the space–time overhead values obtained in ref. [28] cannot directly be compared with those of Table 1. Roughly speaking, one would need to multiply the numbers obtained in ref. [28] by at least six (where the factor of six comes from the fact that six time steps are required to measure all the surface code stabilizers). To be clear, Eq. (6) takes into account the total number of time steps required for each measurement cycle. As such, our scheme provides a space–time overhead improvement, say to obtain a magic state with $p_L^{(5)} = 3.6\times10^{-8}$, by at least an order of magnitude. One of the main reasons for the reduction in overhead is that we did not need to use encoded Clifford operations due to the fault-tolerant properties of our circuits. In ref. [28], the magic state being distilled needed to be encoded in the $d=7$ surface code (so that the logical error rates of the encoded Clifford gates are ~10^{-9}) to obtain an output state with failure probability 4.4×10^{-8}. On the other hand, our scheme does not require encoded Clifford gates with very low error rates, and instead works with physical Clifford gates with an error rate $p=10^{-4}$.

Performance of the scheme with encoded Clifford operations

While $|\overline{H}\rangle$ states with logical error rates near 10^{-8} and 10^{-10} may be sufficient for certain quantum chemistry applications[50], $|\overline{H}\rangle$ states with even lower logical error rates are required for many quantum algorithms. To achieve such low error rates, all stabilizer operations in our magic state preparation schemes in Fig. 1 need to be encoded in an error-correcting code so that they have much lower error rates compared to unencoded stabilizer operations. We now discuss the implementation of our $|\overline{H}\rangle$ preparation scheme presented in Fig. 1 with encoded stabilizer operations. In what follows, we denote $|\overline{H}\rangle_f$ as the state produced when implementing the scheme of Fig. 1 with encoded stabilizer operations. Further, the circuits $G^{(1\to d)}$, $H_m^{(d)}$ and $EC^{(d)}$ will be denoted as $\tilde{G}^{(1\to d)}$, $\tilde{H}_m^{(d)}$ and $\tilde{EC}^{(d)}$ when implemented with encoded stabilizer operations.

Suppose that in order to prepare an $|\overline{H}\rangle_f$ state with some target logical error rate, we require all stabilizer operations to be encoded in a triangular color code of distance d_2. First, $|\overline{H}\rangle_{d_1}$ states are prepared using the scheme described in Fig. 1 (with $d_1 \in \{3, 5, 7\}$ and with physical stabilizer operations). The distance d_1 is chosen such that the $|\overline{H}\rangle_{d_1}$ states have smaller logical failure rates compared with those of the distance d_2 encoded stabilizer operations. The $|\overline{H}\rangle_{d_1}$ states are used to implement the logical T gates (see Fig. 5a) and for injection in the circuit $\tilde{G}^{(1\to d_f)}$. If $d_1 < d_2$,

Table 2. Performance of our magic state preparation scheme using stabilizer operations encoded in the color code.

Outer code distance	Code distance for $|\overline{H}\rangle_{d_1}$	Physical error rate p	Color code distance	Logical error rate $p_L^{(d_f)}$	Average qubit overhead	Minimum qubit overhead	Space–time overhead
$d_f = 3$	$d_1 = 7$	10^{-4}	$d_2 = 11$	3.59×10^{-15}	10,917	6288	3.91×10^{6}
$d_f = 5$	$d_1 = 5$	10^{-4}	$d_2 = 9$	1.10×10^{-17}	14,955	12,795	1.22×10^{7}
$d_f = 5$	$d_1 = 5$	2×10^{-4}	$d_2 = 11$	4.01×10^{-16}	25,265	19,320	1.88×10^{7}
$d_f = 5$	$d_1 = 5$	3×10^{-4}	$d_2 = 15$	6.12×10^{-17}	53,449	36,420	3.84×10^{7}
$d_f = 7$	$d_1 = 3$	10^{-4}	$d_2 = 7$	5.37×10^{-18}	16,262	15,600	2.30×10^{7}
$d_f = 7$	$d_1 = 3$	2×10^{-4}	$d_2 = 9$	5.11×10^{-17}	28,141	26,364	3.91×10^{7}
$d_f = 7$	$d_1 = 3$	3×10^{-4}	$d_2 = 11$	9.87×10^{-17}	43,329	39,936	5.96×10^{7}
$d_f = 7$	$d_1 = 3$	4×10^{-4}	$d_2 = 13$	5.74×10^{-16}	62,540	56,316	9.26×10^{7}
$d_f = 7$	$d_1 = 3$	5×10^{-4}	$d_2 = 15$	1.17×10^{-15}	84,200	75,504	1.156×10^{8}

Qubit and space–time overhead of various schemes using encoded Clifford gates to obtain $|\overline{H}\rangle_f$ states with logical error rates $p_L^{(d_f)} < 4\times10^{-15}$. Here d_2 is the triangular color code distance used to encode the logical Clifford gates, d_f is distance used for the $|\overline{H}\rangle$ state preparation scheme of Fig. 1, d_1 is the distance of $|\overline{H}\rangle$ prior to being grown into $|\overline{H}\rangle_{d_2}$ and p is the physical error rate. See Supplementary Note 5 for a detailed description of how to estimate the resource overhead. See also Supplementary Table 1.

Table 3. Performance of our magic state preparation scheme using stabilizer operations encoded in the surface code.

Outer code distance	Code distance for $|\overline{H}\rangle_{d_1}$	Physical error rate p	Surface code distance	Logical error rate $p_L^{(d_f)}$	Average qubit overhead	Minimum qubit overhead
$d_f = 7$	$d_1 = 3$	10^{-4}	$d_2 = 5$	5.07×10^{-17}	10,025	9506
$d_f = 7$	$d_1 = 3$	10^{-3}	$d_2 = 11$	8.11×10^{-20}	60,886	47,324

Qubit overhead of schemes for obtaining an encoded $|\overline{H}\rangle_f$ as in Table 2, but with logical stabilizer operations encoded in a distance d_2 surface code. Note that the states $|\overline{H}\rangle_{d_2}$ are first encoded in the color code, and lattice surgery is performed to obtain an $|\overline{H}\rangle_{d_2}$ state encoded in the surface code as in ref. [52] . See Supplementary Note 5 for a detailed description of how to estimate the resource overhead costs of our scheme. See also Supplementary Table 2.

the $|\overline{H}\rangle_{d_1}$ states must first be grown into $|\overline{H}\rangle_{d_2}$ states using the growing circuit $G^{(d_1\to d_2)}$ shown in Fig. 7c. Note that we did not put the tilde on $G^{(d_1\to d_2)}$ since all operations are implemented with physical gates (see also the Methods section). Using encoded $|\overline{H}\rangle_{d_2}$ states ensures that stabilizer operations encoded in a distance d_2 triangular color code can be used with the prepared magic states. See Supplementary Note 5 and Supplementary Figs. 6 and 7 for a more detailed description of the setting and how to estimate the resource overhead.

The numerical values obtained in Table 2 shows that the least costly scheme to prepare the state $|\overline{H}\rangle_f$ with a logical error rate $p_L^{(d_f)}<4\times10^{-15}$ when $p = 10^{-4}$ is to first prepare the states $|\overline{H}\rangle_{d_1}$ with $d_1 = 7$, and to grow these states to encoded $d_2 = 11$ states. Using distance $d_2 = 11$ encoded stabilizer operations, the final magic state $|\overline{H}\rangle_f$ is prepared using the distance $d_f = 3$ scheme in Fig. 1. On average, the amount of qubits required to prepare such a state is 10,917 and the space–time overhead is 3.91×10^{6}. If the time cost for preparing the $|\overline{H}\rangle_{d_1}$ states is of a lesser concern, the minimum number of qubits required to prepare $|\overline{H}\rangle_f$ with $p_L^{(d_f)}<4\times10^{-15}$ is 6288. To compare with other schemes, in ref. [28], a magic state with a logical error failure rate of 2.4×10^{-15} required a minimum of 16,400 qubits.

To obtain a state $|\overline{H}\rangle_f$ with $p_L^{(d_f)}\approx10^{-15}$ when $p = 10^{-3}$ using a small amount of resources requires encoded stabilizer operations with much lower logical failure rates than what is achieved with the triangular color code family. One viable option is to use stabilizer operations encoded in the surface code[51] due to the low error rates that can be achieved when $p = 10^{-3}$. However, in such a setting, after the states $|\overline{H}\rangle_{d_2}$ have been prepared, they must be teleported to the surface code before they can be injected in the circuit of Fig. 5 and in the circuit implementing $\tilde{G}^{(1\to d_f)}$. In particular, one can convert the color code encoded state to the surface code using lattice surgery techniques as was done in ref. [52]. In Table 3, we provide estimates of the qubit overhead for preparing $|\overline{H}\rangle_f$ when the stabilizer operations are encoded in the surface code. We remark that a complete analysis of the resource overhead cost of our scheme using encoded stabilizer operations would require the inclusion of routing costs. In this work, we do not take into account costs associated with the implementation of lattice surgery for performing gates and the transportation of logical qubits. Given that such considerations highly depend on the underlying hardware architecture, we leave such an analysis to future work. In Supplementary Note 5, we discuss alternative schemes with the surface code that do not require lattice surgery techniques.

DISCUSSION

In this work we showed how to prepare an $|H\rangle$-type magic state in a fault-tolerant way using a two-dimensional color code architecture requiring only nearest-neighbor interactions. The proposed architecture can be used to both measure local stabilizers of the color code in addition to a global operator, without the need for changing the qubit layout. Such an architecture was made possible with the use of flag qubits, in addition to an important concept that we introduce and call redundant ancilla encoding. Estimating the performance of our scheme, we showed that when $p = 10^{-4}$, only 68 and 231 qubits are required to prepare an encoded $|H\rangle$ state with logical error rates 3.6×10^{-8} and 4.9×10^{-10}, respectively. In addition, we also showed how our scheme can be used with encoded stabilizer operations to achieve significantly lower logical failure rates, both in the regime where $p=10^{-4}$ and $p=10^{-3}$. We stress that our results were obtained by considering a full circuit-level depolarizing noise model, where all stabilizer operations could fail.

We also point out key differences between magic state distillation schemes and our fault-tolerant methods for preparing magic states. Magic state distillation is a top-down approach, where error detection circuits are designed to prepare magic states without regard to their fault-tolerant properties. It is assumed that appropriate code distances for the encoded Clifford operations will be chosen to ensure that consecutive rounds of distillation will produce higher fidelity magic states. Our scheme, which uses flag-qubits and redundant ancilla encoding, is a bottom-up approach. The fault-tolerant properties of the magic state preparation circuits are prioritized and emphasis is given towards constructing v-flag circuits for large v. As such, to obtain magic states with very low failure rates, if encoded stabilizer operations are required, such operations can fail with error rates commensurate to the magic states being injected.

Note that when considering the implementation of our scheme with encoded stabilizer operations using the surface code, the $d_f=7$ version of our scheme was optimal for both $p=10^{-4}$ and $p=10^{-3}$. A clear direction of future work would be to find a v-flag circuit (with $v\geq 4$) allowing a fault-tolerant implementation of a $d_f \geq 9$ scheme. Such a scheme could potentially further reduce the overhead for preparing $|H\rangle$ states with very low error rates. Moreover when using the surface code for the encoded stabilizer operations, instead of performing lattice surgery to convert a color code encoded $|\overline{H}\rangle_{d_2}$ state to one encoded in the surface code (as we did above), another option would be to initially prepare an encoded $|\overline{H}\rangle_{d_2}$ state in a small distance surface code using some other method, such as a magic state distillation protocol. While these techniques can potentially further reduce the required resource overhead, a careful analysis of the overhead would require choosing the appropriate magic state distillation protocol (or some other scheme that uses fault-tolerant circuits to prepare encoded magic states). Therefore such an analysis is left for future work.

With a plethora of possible research directions for preparing magic states in a fault-tolerant way which build upon our work, and given the realistic hardware constraints that are built directly into our schemes, we believe our work paves the way for achieving very low overhead universal quantum computation with both near-term and long-term quantum devices.

METHODS

H-type magic states

The n-qubit Clifford group is defined as

$$\mathcal{P}_n^{(2)} = \{U : \forall P \in \mathcal{P}_n^{(1)}, UPU^{\dagger} \in \mathcal{P}_n^{(1)}\}, \tag{7}$$

where $\mathcal{P}_n^{(1)}$ is the n-qubit Pauli group. The Clifford group is generated by

$$\mathcal{P}_n^{(2)} = \langle H, Y\left(\frac{\pi}{2}\right), CNOT\rangle, \tag{8}$$

where

$$H = \frac{1}{\sqrt{2}}\left(111-1\right), \text{ and } Y\left(\tfrac{\pi}{2}\right) = \tfrac{1}{\sqrt{2}}\left(1-111\right). \tag{9}$$

Here H is the Hadamard gate, $Y\left(\frac{\pi}{2}\right)=e^{-i\frac{\pi}{4}Y}$ and the CNOT gates acts as

$$\text{CNOT}|a\rangle \otimes |b\rangle = |a\rangle \otimes |a \oplus b\rangle, \tag{10}$$

on the computational basis states $|a\rangle$ and $|b\rangle$. The Clifford group, along with the non-Clifford gate

$$T = e^{-i\frac{\pi Y}{8}} = \left(\cos\tfrac{\pi}{8} - \sin\tfrac{\pi}{8}\sin\tfrac{\pi}{8}\cos\tfrac{\pi}{8}\right), \tag{11}$$

forms a universal gate set. That is, defining $\mathcal{G}=\langle H, Y\left(\frac{\pi}{2}\right), T, CNOT\rangle$ and given a target fidelity ϵ, a unitary operator U can be approximated with $\mathcal{O}(\log^c 1/\epsilon)$ gates in $\mathcal{G}$[52,53]. Note that the T gate defined in Eq. (11) is Clifford equivalent to the gate $\tilde{T}=diag(1, e^{i\pi/4})$, which is also commonly referred to as the T gate. Whenever we refer to T gates throughout this paper, they will always correspond to the gate given in Eq. (11).

A magic state is a state that can be used as a resource state to simulate non-Clifford gates using only stabilizer operations (i.e., Clifford gates, computational basis states and Z-basis measurements). Additionally, magic states can also be distilled using only stabilizer operations[18]. In this paper we focus entirely on preparing an $|H\rangle$-type magic state[18,20]. In particular, an $|H\rangle$ state is given by

$$|H\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle = T|0\rangle, \tag{12}$$

which is a $+1$ eigenstate of H. Note that $|H\rangle$ is Clifford equivalent to the following more commonly known magic state

$$|A_{\frac{\pi}{4}}\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle) = e^{i\frac{\pi}{8}}HS^{\dagger}|H\rangle, \tag{13}$$

where $S=diag(1,\ i)$ is the phase gate. The state $|A_{\frac{\pi}{4}}\rangle$ can be used to simulate the $\tilde{T}=diag(1, e^{i\pi/4})$ gate using stabilizer operations.

In Fig. 5a, we provide the circuit used to simulate the T gate in Eq. (11) using one $|H\rangle$ state in addition to stabilizer operations. In many physical implementations however, noisy $|H\rangle$ states are injected into such circuits. To determine if an $|H\rangle$ state is afflicted by an error, one can measure the Hadamard operator using the circuit show in Fig. 5b. Since $HY|H\rangle=-Y|H\rangle$, if $|H\rangle$ is afflicted by a Y error, a -1 measurement outcome will be obtained. Further, if $|H\rangle$ is afflicted by an X or Z error, then the measurement outcome is ± 1 at random. If the outcome is $+1$, then no error will be present after the measurement. Note that the controlled-Hadamard gate can be decomposed into products of T, $T^{\dagger}$ and controlled-Z gates as shown in Fig. 5c.

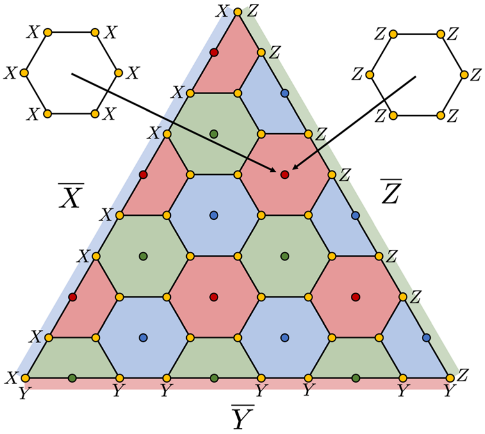

Fig. 6 Triangular color code. Lattice $\mathcal{L}$ for the implementation of the triangular color code (in this case a distance $d=5$ color code). Each face of $\mathcal{L}$ consists of both X and Z-type stabilizer generators which are supported on all qubits belonging to the face. The logical X and Z operators are given by tensor products of all X and all Z operators along a boundary of the triangle.

Triangular color codes

Color codes are topological codes, and thus the data qubits can placed on a lattice where each stabilizer generator can be measured using nearest-neighbor interactions. The triangular color code family has code parameters $[[n=(3d^2+1)/4,\ 1,\ d]]$ and is a version of the color code defined on a two-dimensional lattice $\mathcal{L}$ with boundaries. It is a self-dual CSS code with weight-four and weight-six X and Z-type stabilizers (see Fig. 6). The lattice $\mathcal{L}$ is 3-colorable, meaning that every face can be colored in red, green, or blue with any other face sharing an incident edge having a different color. All vertices of $\mathcal{L}$ (apart from the three corners) are incident to three edges. The distance-d triangular color code can correct any errors of weight less than or equal to $(d-1)/2$ where d is odd. Further, triangular color codes can implement all logical Clifford gates using transversal operations. In particular, the logical Hadamard operator is simply given by $\overline{H}=H^{\otimes n}$ where n is the number of data qubits.

In ref. [42], an efficient decoder (which we refer to as the Lift decoder) for two-dimensional color codes was provided. In ref. [38] it was shown how the Lift decoder can be extended to color codes with boundaries. Further, it

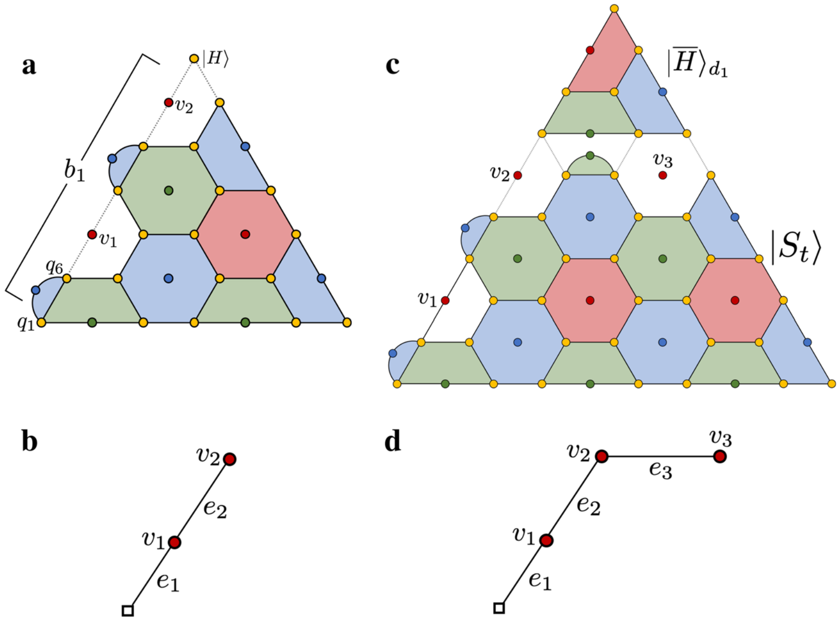

Fig. 7 Growing the triangular color code. a Circuit $G^{(1\to5)}$ for growing a physical $|H\rangle$ state to a distance $d=5$ color code. First, a stabilizer state $|S_t\rangle$ is prepared with stabilizers given by the red, green and blue plaquettes. The encoded $|\overline{H}\rangle$ state is then obtained by measuring both weight-four X followed by Z-type operators (represented by white plaquettes) along the boundary of the triangle. Such measurements are random since the weight-four operators do not commute with the weight-two stabilizers of $|S_t\rangle$. **b** Matching graph (denoted by $G^{(5)}_{1x}$ and $G^{(5)}_{1z}$) used to implement weight-two corrections arising from -1 measurement outcomes for generators of $\mathcal{S}_{b_1}$. As an example, if the vertex v_1 is highlighted, after implementing minimum-weight perfect matching (MWPM), the edge e_1 would be selected resulting in the correction $Z_{q_1}Z_{q_6}$. **c** Circuit $G^{(3\to7)}$ for growing an $|\overline{H}\rangle$ state encoded in a $d=3$ triangular color code to one encoded in a $d=7$ triangular color code. A stabilizer state $|S_t\rangle$ is first prepared, and the $d=3$ $|\overline{H}\rangle$ state is prepared, e.g., following the methods of Fig. 1. One then measures the X and Z-type operators supported on the white plaquettes which anti-commute with the weight-two generators of $\mathcal{S}_{st}$ resulting in random measurement outcomes. The measurements of all plaquettes are repeated three times to distinguish measurement errors from random measurement outcomes and to correct errors. **d** Matching graph (denoted by $G^{(7)}_{3x}$ and $G^{(7)}_{3z}$) used to implement the weight-two corrections arising from random measurement outcomes of the operators supported on the white plaquettes. As in **b**, each edge corresponds to two qubits supported on a weight-two generator of $\mathcal{S}_{st}$.

was shown how the Lift decoder can incorporate measurement outcomes from flag qubits to maintain the effective distance of the code under a full circuit-level noise model (see below). Using such methods, it was found that triangular color codes exhibit a competitive threshold value of 0.2% under a full circuit-level depolarizing noise model.

Growing the triangular color code

An illustration of the circuit $G^{(1\to5)}$ for growing a physical $|H\rangle$ state to a logical $|\overline{H}\rangle$ state encoded in the $d=5$ triangular color code is given in Fig. 7a. The preparation of $|\overline{H}\rangle$ in a general distance d triangular color code can be done as follows. First, one prepares a stabilizer state $|S_t\rangle$ (which encodes no logical qubits) that is stabilized by all elements in $\mathcal{S}_{st}=\mathcal{S}_{w_2}\cup(\mathcal{S}_{\text{color}}\setminus\mathcal{S}_{b_1})$ where $\mathcal{S}_{\text{color}}$ is the stabilizer group of a distance d triangular color code, $\mathcal{S}_{b_1}$ is generated by the X and Z-type weight-four operators (white plaquettes) along the boundary b_1 of the triangular color code and $\mathcal{S}_{w_2}$ is generated by the weight-two X and Z-type operators along the boundary b_1 (see Fig. 7 for the case where $d=5$). The qubit that is in $\text{supp}(\mathcal{S}_{\text{color}})\setminus\text{supp}(\mathcal{S}_{\text{st}})$ is prepared in the physical $|H\rangle$ state. Note that in the above construction, any of the three boundaries of $\mathcal{L}$ can be chosen. We chose b_1 for convention.

After preparing $|S_t\rangle$, the X and Z-type generators of $\mathcal{S}_{b_1}$ are measured along the boundary b_1. Since the generators of $\mathcal{S}_{b_1}$ do not commute with the weight-two generators of $\mathcal{S}_{w_2}$, the measurement outcomes of each generator in $\mathcal{S}_{b_1}$ will be ±1 at random. If a -1 outcome is obtained, a Pauli frame update[54–57] needs to be applied to obtain the correct encoded state. In order to perform the correct Pauli frame update based on the random measurement outcomes, we define two one-dimensional graphs $G^{(d)}_{1x}$ and $G^{(d)}_{1z}$. The vertices of the graph $G^{(d)}_{1x}$ contain the random measurement outcomes of the X-type generators in $\mathcal{S}_{b_1}$. Each edge corresponds to two qubits that have support on one of the weight-two operators of $\mathcal{S}_{w_2}$. Similarly, the vertices of the graph $G^{(d)}_{1z}$ encodes the random measurement outcomes of the Z-type generators in $\mathcal{S}_{b_1}$. An example is provided in Fig. 7a for the $d=5$ triangular color code. Given the set of highlighted vertices of $G^{(d)}_{1x}$ and $G^{(d)}_{1z}$, Minimum-Weight-Perfect-Matching (MWPM)[58] is applied on both graphs. Each highlighted edge involves performing a weight-two X or Z-type Pauli frame update. For instance, for the graph $G^{(5)}_{1x}$ in Fig. 7b, if the edge e_1 is selected during MWPM, the correction $Z_{q_1}Z_{q_6}$ is applied to the data. Note that the growing scheme is not fault-tolerant, and thus there is no need to repeat the measurements described above.

Growing an $|\overline{H}\rangle$ state encoded in a distance-d_1 triangular color code to one encoded in a distance-d_2 triangular color code (with $d_2>d_1$) can be done similarly as above and we denote such a growing circuit by $G^{(d_1\to d_2)}$. The circuit $G^{(d_1\to d_2)}$ is implemented as follows. As in the case of $G^{(1\to d)}$, a stabilizer state $|S_t\rangle$ is prepared, and operators supported on the white plaquettes seperating $|\overline{H}\rangle_{d_1}$ and $|S_t\rangle$ are measured (this step can be viewed as gauge fixing of an underlying subsystem code[59]). Measurements of all operators supported on each plaquette of the distance d_2 triangular color code are repeated d_1 times to correct errors and to distinguish measurement errors from the random outcomes obtained when measuring the white plaquettes. An example for implementing $G^{(3\to7)}$ is provided in Fig. 7c and d. More details on the implementation of the growing circuits $G^{(1\to d)}$ and $G^{(d_1\to d_2)}$ are provided in Supplementary Note 1 and Supplementary Fig. 1.

Circuit-level noise model and simulation details

The full circuit-level noise model used throughout all simulations performed in this work with physical stabilizer operations is given as follows:

- With probability p, each single-qubit gate location is followed by a Pauli error drawn uniformly and independently from $\{X, Y, Z\}$.
- With probability p, each two-qubit gate is followed by a two-qubit Pauli error drawn uniformly and independently from $\{I, X, Y, Z\}^{\otimes 2} \setminus \{I \otimes I\}$.
- With probability $\frac{2p}{3}$, the preparation of the $|0\rangle$ state is replaced by $|1\rangle = X|0\rangle$. Similarly, with probability $\frac{2p}{3}$, the preparation of the $|+\rangle$ state is replaced by $|-\rangle = Z|+\rangle$.
- With probability p, the preparation of the $|H\rangle$ state is replaced by $P|H\rangle$ where P is a Pauli error drawn uniformly and independently from $\{X, Y, Z\}$.
- With probability $\frac{2p}{3}$, any single qubit measurement has its outcome flipped.
- With probability p, each idle gate location is followed by a Pauli error drawn uniformly and independently from $\{X, Y, Z\}$.

It is important to point out that due to the presence of T gates (which are non-Clifford) used to implement the controlled-Hadamard gates (see Fig. 5), an efficient Monte–Carlo simulation of the circuits $H_m^{(d)}$ using Gottesman–Knill error propagation[60,61] is not possible. Consequently, we divided the Monte–Carlo simulation to calculate $p_L^{(d)}$ and $p_{\text{acc}}^{(d)}(p)$ into two parts. First, we perform a Monte–Carlo simulation of the circuit used to prepare $|\overline{H}\rangle_G$. If the output error E_{out} has a non-trivial syndrome or is a logical operator, the protocol of Fig. 1 is aborted, otherwise, we proceed to simulate the $H_m^{(d)}$ and $EC^{(d)}$ circuits. We define $p_{\text{acc},1}^{(d)}$ to be the probability of proceeding to the $H_m^{(d)}$ and $EC^{(d)}$ circuits. Note that there could be other faults in the $H_m^{(d)}$ and $EC^{(d)}$ circuits that would cause the protocol to be accepted even though E_{out} had a non-trivial syndrome or was a logical operator. However such an event would require at least $(d-1)/2$ faults, and since the large majority of malignant error locations are found in $H_m^{(d)}$ and $EC^{(d)}$, such an approximation only affects α in Eq. (1) by a small constant factor. As an example for $p = 10^{-4}$, $p_{\text{acc},1}^{(d)} = 0.9897(d=3)$, $p_{\text{acc},1}^{(d)} = 0.9711(d=5)$, and $p_{\text{acc},1}^{(d)} = 0.9462(d=7)$. As such, one can see that these probabilities are close to 1 for all relevant code distances. Furthermore, we numerically verify that the probability of a logical failure when preparing $|\overline{H}\rangle_G$ non fault-tolerantly is $< 90p$ for $d \le 7$, hence, $< 1\%$ for $p = 10^{-4}$. Note that these logical errors will be detected in the later stages of Fig. 1 unless there are strictly more than $(d-1)/2$ faults in the entire protocol, which is very unlikely.

If all flag qubit and ancilla qubit measurement outcomes in the $(d-1)/2$ applications of the $H_m^{(d)}$ and $EC^{(d)}$ circuits are trivial, the output state is accepted. We define $p_{\text{acc},2}^{(d)}$ to be the probability of acceptance for the second part of the simulation (i.e., the simulation of the $H_m^{(d)}$ and $EC^{(d)}$ circuits). Hence the total acceptance probability is $p_{\text{acc}}^{(d)} = p_{\text{acc},1}^{(d)} p_{\text{acc},2}^{(d)}$. To determine if the output error of an accepted state is correctable, we perform one round of perfect error correction using the the Lift decoder.

We also remark that to simulate the circuit $H_m^{(d)}$, we use the fact that $T^\dagger X T = \frac{1}{\sqrt{2}}(X+Z) = H$ and $T^\dagger Z T = \frac{1}{\sqrt{2}}(Z-X) = iYH$. Hence if an X or Z error is input to a T or $T^\dagger$ gate, we pessimistically apply an X or Z error to the output, each with 50% probability. Such an approximation would be exact if twirling operations were performed both before and after the $T^\dagger$ and T gates (see Supplementary Note 3 and Supplementary Fig. 3 for more details; see also[9]). To be clear, we do not propose applying twirling operations when implementing our scheme as this could reduce the performance. The approximations stated here are performed to allow us to simulate our scheme on a classical computer. More details on the simulation of non-Clifford gates are provided in Supplementary Note 4 and Supplementary Figs. 4 and 5. Note that instead of twirling, one could possibly perform an exact simulation using methods introduced in refs. [62,63]. We leave such a more sophisticated simulation as a future research direction.

DATA AVAILABILITY

All the data used to generate the results of our work can be found in the public repository https://github.com/einsteinchris/MagicStatePrep.git under the file name DataMagicStatePrepSim.nb.

CODE AVAILABILITY

The code used to generate the data in this work cannot be shared due to proprietary reasons.

Received: 30 April 2020; Accepted: 10 September 2020;
Published online: 27 October 2020

REFERENCES

1. Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. 35th Annual Symposium on Foundations of Computer Science*, 124–134 (IEEE, 1994).
2. Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).
3. Eastin, B. & Knill, E. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.* **102**, 110502 (2009).
4. Knill, E., Laflamme, R. & Zurek, W. H. Threshold accuracy for quantum computation. Preprint at https://arxiv.org/abs/quant-ph/9610011 (1996).
5. Paetznick, A. & Reichardt, B. W. Universal fault-tolerant quantum computation with only transversal gates and error correction. *Phys. Rev. Lett.* **111**, 090505 (2013).
6. Jochym-O'Connor, T. & Laflamme, R. Using concatenated quantum codes for universal fault-tolerant quantum gates. *Phys. Rev. Lett.* **112**, 010505 (2014).
7. Anderson, J. T., Duclos-Cianci, G. & Poulin, D. *Phys. Rev. Lett.* **113**, 080501 (2014).
8. Bombín, H. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *N. J. Phys.* **17**, 083002 (2015).
9. Bravyi, S. & Cross, A. Doubled color codes. Preprint at https://arxiv.org/abs/1509.03239 (2015).
10. Jochym-O'Connor, T. & Bartlett, S. D. Stacked codes: universal fault-tolerant quantum computation in a two-dimensional layout. *Phys. Rev. A* **93**, 022323 (2016).
11. Yoder, T. J., Takagi, R. & Chuang, I. L. Universal fault-tolerant gates on concatenated stabilizer codes. *Phys. Rev. X* **6**, 031039 (2016).
12. Chamberland, C., Jochym-O'Connor, T. & Laflamme, R. Thresholds for universal concatenated quantum codes. *Phys. Rev. Lett.* **117**, 010501 (2016).
13. Chamberland, C., Jochym-O'Connor, T. & Laflamme, R. Overhead analysis of universal concatenated quantum codes. *Phys. Rev. A* **95**, 022313 (2017).
14. Chamberland, C. & Jochym-O'Connor, T. Error suppression via complementary gauge choices in reed-muller codes. *Quantum Sci. Technol.* **2**, 035008 (2017).
15. Chamberland, C. & Cross, A. W. Fault-tolerant magic state preparation with flag qubits. *Quantum* **3**, 143 (2019).
16. Jochym-O'Connor, T. Fault-tolerant gates via homological product codes. *Quantum* **3**, 120 (2019).
17. Lavasani, A., Zhu, G. & Barkeshli, M. Universal logical gates with constant overhead: instantaneous Dehn twists for hyperbolic quantum codes. *Quantum* **3**, 180 (2019).
18. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A* **71**, 022316 (2005).
19. Reichardt, B. W. Quantum universality from magic states distillation applied to css codes. *Quantum Inf. Process.* **4**, 251–264 (2005).
20. Meier, A. M., Eastin, B. & Knill, E. Magic-state distillation with the four-qubit code. *Quantum Info Comput.* **13**, 0195–0209 (2013).
21. Bravyi, S. & Haah, J. Magic-state distillation with low overhead. *Phys. Rev. A* **86**, 052329 (2012).
22. Fowler, A. G., Devitt, S. J. & Jones, C. Surface code implementation of block code state distillation. *Sci. Rep.* **1939**, 2045–2322 (2013).
23. O'Gorman, J. & Campbell, E. T. Quantum computation with realistic magic-state factories. *Phys. Rev. A* **95**, 032338 (2017).
24. Haah, J., Hastings, M. B., Poulin, D. & Wecker, D. Magic state distillation with low space overhead and optimal asymptotic input count. *Quantum* **1**, 31 (2017).
25. Haah, J., B. Hastings, M., Poulin, D. & Wecker, D. Magic state distillation at intermediate size. *Quantum Info Comput.* **18**, 97–165 (2018).
26. Hastings, M. B. & Haah, J. Distillation with sublogarithmic overhead. *Phys. Rev. Lett.* **120**, 050504 (2018).
27. Haah, J. & Hastings, M. B. Codes and protocols for distilling *T*, controlled-*S*, and toffoli gates. *Quantum* **2**, 71 (2018).
28. Litinski, D. Magic state distillation: not as costly as you think. *Quantum* **3**, 205 (2019).
29. Yoder, T. J. & Kim, I. H. The surface code with a twist. *Quantum* **1**, 2 (2017).
30. Chao, R. & Reichardt, B. W. Quantum error correction with only two extra qubits. *Phys. Rev. Lett.* **121**, 050502 (2018).
31. Chao, R. & Reichardt, B. W. Fault-tolerant quantum computation with few qubits. *npj Quantum Inf.* **4**, 42 (2018).
32. Chamberland, C. & Beverland, M. E. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum* **2**, 53 (2018).

33. Tansuwannont, T., Chamberland, C. & Leung, D. Flag fault-tolerant error correction, measurement, and quantum computation for cyclic calderbank-shor-steane codes. *Phys. Rev. A* **101**, 012342 (2020).
34. Reichardt, B. W. Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits. Preprint at https://arxiv.org/abs/1804.06995 (2018).
35. Shi, Y., Chamberland, C. & Cross, A. Fault-tolerant preparation of approximate GKP states. *N. J. Phys.* **21**, 093007 (2019).
36. Chamberland, C., Zhu, G., Yoder, T. J., Hertzberg, J. B. & Cross, A. W. Topological and subsystem codes on low-degree graphs with flag qubits. *Phys. Rev. X* **10**, 011022 (2020).
37. Chao, R. & Reichardt, B. W. Flag fault-tolerant error correction for any stabilizer code. *PRX Quantum* **1**, 010302 (2020).
38. Chamberland, C., Kubica, A., Yoder, T. J. & Zhu, G. Triangular color codes on trivalent graphs with flag qubits. *N. J. Phys.* **22**, 023019 (2020).
39. Bombin, H. & Martin-Delgado, M. A. Topological quantum distillation. *Phys. Rev. Lett.* **97**, 180501 (2006).
40. Kubica, A. & Beverland, M. E. Universal transversal gates with color codes: A simplified approach. *Phys. Rev. A* **91**, 032330 (2015).
41. Kubica, A. The ABCs of the color code: a study of topological quantum codes as toy models for fault-tolerant quantum computation and quantum phases of matter. Ph.D. thesis (2018). https://thesis.library.caltech.edu/10955/.
42. Kubica, A. & Delfosse, N. Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders. Preprint at https://arxiv.org/abs/1905.07393.
43. Fowler, A. G., Mariantoni, M., Martinis, J. M. & Cleland, A. N. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A* **86**, 032324 (2012).
44. Kivlichan, I. D. et al. Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization. *Quantum* **4**, 296 (2020).
45. Gidney, C. & Ekerå, M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Preprint at https://arxiv.org/abs/1905.09749 (2019).
46. Aliferis, P., Gottesman, D. & Preskill, J. Quantum accuracy threshold for concatenated distance-3 codes. *Quantum Info Comput.* **6**, 97–165 (2006).
47. Gottesman, D. An introduction to quantum error correction and fault-tolerant quantum computation. *Proc. Symp. Appl. Math.* **68**, 13–58 (2010).
48. Campbell, E. T., Terhal, B. M. & Vuillot, C. Roads towards fault-tolerant universal quantum computation. *Nature* **549**, 172–179 (2017).
49. Delfosse, N. Hierarchical decoding to reduce hardware requirements for quantum computing. Preprint at https://arxiv.org/abs/2001.11427 (2019).
50. McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C. & Yuan, X. Quantum computational chemistry. *Rev. Mod. Phys.* **92**, 015003 (2020).
51. Fowler, A. G., Whiteside, A. C., McInnes, A. L. & Rabbani, A. Topological code autotune. *Phys. Rev. X* **2**, 041003 (2012).
52. Kitaev, A. Y. Quantum computations: algorithms and error correction. *Russ. Math. Surv.* **52**, 1191–1249 (1997).
53. Dawson, C. M. & Nielsen, M. The solovay-kitaev algorithm. *Quant. Inf. Comput.* **6**, 81–95 (2006).
54. Knill, E. Quantum computing with realistically noisy devices. *Nature* **434**, 39–44 (2005).
55. Terhal, B. M. Quantum error correction for quantum memories. *Rev. Mod. Phys.* **87**, 307–346 (2015).
56. DiVincenzo, D. P. & Aliferis, P. Effective fault-tolerant quantum computation with slow measurements. *Phys. Rev. Lett.* **98**, 020501 (2007).
57. Chamberland, C., Iyer, P. & Poulin, D. Fault-tolerant quantum computing in the pauli or clifford frame with slow error diagnostics. *Quantum* **2**, 43 (2018).
58. Edmonds, J. Paths, trees, and flowers. *Can. J. Math.* **17**, 449–467 (1965).
59. Vuillot, C. et al. Code deformation and lattice surgery are gauge fixing. *N. J. Phys.* **21**, 033028 (2019).
60. Gottesman, D. The Heisenberg representation of quantum computers. In: *Proc. XXII International Colloquium on Group Theoretical Methods in Physics* 32–43 (Cambrisdge, Mass.: International Press, 1998).
61. Aaronson, S. & Gottesman, D. Improved simulation of stabilizer circuits. *Phys. Rev. A* **70**, 052328 (2004).
62. Bravyi, S. & Gosset, D. Improved classical simulation of quantum circuits dominated by clifford gates. *Phys. Rev. Lett.* **116**, 250501 (2016).
63. Bravyi, S. et al. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum* **3**, 181 (2019).

ACKNOWLEDGEMENTS

C.C. acknowledges Yihe Tang and Milan Cvitkovic for their help in setting up the computing resources with the AWS clusters that were used for performing all the numerical simulations in this work. We would like to thank Earl Campbell, Aleksander Kubica and Fernando Brandao for useful discussions. We thank Simone Severini, Bill Vass and Dominique L'Eplattenier for their guidance and help with submitting the paper. We also thank Kevin Dothager for his help with IP.

AUTHOR CONTRIBUTIONS

C.C. and K.N. conceived the idea of redundant ancilla encoding, designed the fault-tolerant circuits for both syndrome extraction and measuring Hadamard, and proved the fault-tolerant properties of the magic state preparation protocol. C.C. performed the numerics and overhead analysis. C.C. and K.N. wrote the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information is available for this paper at https://doi.org/10.1038/s41534-020-00319-5.

Correspondence and requests for materials should be addressed to C.C. or K.N.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.