Low-depth random Clifford circuits for quantum coding against Pauli noise using a tensor-network decoder

Andrew S. Darmawan,[1,2,*] Yoshifumi Nakata,[1] Shiro Tamiya,[3] and Hayata Yamasaki[4,2]

[1] *Yukawa Institute of Theoretical Physics (YITP), Kyoto University, Kitashirakawa Oiwakecho, Sakyo-ku, Kyoto 606-8502, Japan*

[2] *JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan*

[3] *Department of Applied Physics, Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo,*
Bunkyo-ku, Tokyo 113-8656, Japan

[4] *Department of Physics, Graduate School of Science, The University of Tokyo, 7-3-1 Hongo,*
Bunkyo-ku, Tokyo 113-0033, Japan

(Received 25 January 2023; accepted 6 March 2024; published 16 April 2024)

Recent work [M. J. Gullans *et al.*, Phys. Rev. X **11**, 031066 (2021)] has shown that quantum error correcting codes defined by random Clifford encoding circuits can achieve a nonzero encoding rate in correcting errors even if the random circuits on n qubits, embedded in one spatial dimension (1D), have a logarithmic depth $d = O(\log n)$. However, this was demonstrated only for a simple erasure noise model. In this work, we discover that, for the same class of codes, this desired property indeed holds for the conventional Pauli noise model. Specifically, we numerically demonstrate that the hashing bound, i.e., a rate known to be achieved with $d = O(n)$-depth random encoding circuits, can be attained for the above codes even when the circuit depth is restricted to $d = O(\log n)$ in 1D for depolarizing noise of various strengths. This analysis is made possible with our development of a tensor-network maximum-likelihood decoding algorithm that works efficiently for log-depth encoding circuits in 1D.

DOI: 10.1103/PhysRevResearch.6.023055

Protecting quantum information from noise and decoherence is a requirement for scalable quantum computation, which, in theory, can be done using quantum error-correcting codes. Yet despite rapid developments in experimental realizations of quantum error-correcting codes [1–8], there are many challenges that must be overcome before they can be used in practical applications.

One major challenge is to deal with the daunting overhead required for various error-correction schemes. For instance, experimental realization of the surface code is being pursued by several groups due to its high threshold and two-dimensional (2D) layout [9,10]. However, its major drawback appears to be a low encoding rate, defined as $r := k/n$, where k and n are the numbers of encoded and physical qubits of a quantum error-correcting code, respectively. For practical computations, thousands of physical qubits may be required for each encoded qubit [11].

As a result, a significant effort has been made to find error-correction schemes with a higher encoding rate and lower overhead. A variety of schemes have been proposed based on quantum low-density parity-check (LDPC) codes [12–20] and concatenated quantum codes [21]. However, the existing high-rate quantum LDPC codes are hard to implement in many architectures since these codes require long-range two-qubit gates. The high-rate concatenated code can suppress errors even if we can only use nearest-neighbor noisy two-qubit gates in a 2D layout [21,22], but it may still need a higher circuit depth than is available under current technologies to attain sufficient error suppression.

Alternative approaches have been proposed, based on random encoding. Random stabilizer codes are known to achieve a nonzero rate with vanishing error probability in the limit of large n for certain types of noise [23–25]. The rate achievable by random stabilizer codes against independently and ideally distributed (IID) Pauli noise $\mathcal{N}(\rho) = p_I \rho + p_X X \rho X + p_Y Y \rho Y + p_Z Z \rho Z$ is known as the hashing bound $r = 1 - H(\vec{p})$, where X, Y, Z are Pauli matrices, $\vec{p} = (p_I, p_X, p_Y, p_Z)$ represent Pauli error probabilities, and H is the Shannon entropy [24]. While the hashing bound is not always optimal, it is relatively high compared to known upper bounds on the optimal rate [26,27]. It was shown by Brown and Fawzi [28,29] that asymptotically the same performance can be achieved by random Clifford encoding circuits even when the circuit depth is only $O(\log^3 n)$.

From a practical perspective, the above random codes have some shortcomings. In particular, an efficient decoding procedure for them is not known and they require all-to-all connectivity (which is not available in many physical architectures). A recent result of Gullans *et al.* [30] has shown that random encoding by Clifford circuits with logarithmic depth and 1D connectivity or sublogarithmic depth in higher dimensions can achieve a nonzero rate against erasure noise.

[*] andrew.darmawan@yukawa.kyoto-u.ac.jp

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

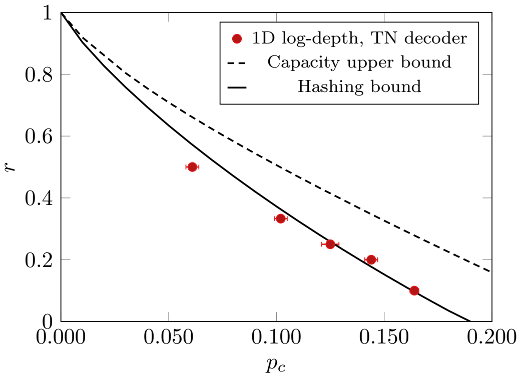

FIG. 1. Numerical results for encoding rate r vs threshold depolarizing probabilities p_c of 1D log-depth circuits using a tensor network decoder (red markers) compared with various analytical bounds. The solid line is the hashing bound, and the dashed line is an upper bound on the capacity, derived in Ref. [27].

A similar performance was observed against erasure noise in an alternative construction of codes, based on constraint satisfaction algorithms [31]. While restricting to the erasure noise model greatly simplifies the decoding of such codes, it involves the strong assumption that the locations of all physical errors are known, which is not the case in current quantum computing architectures. Hence, to be practically relevant, it is necessary to investigate the performance of such codes against more realistic noise models.

In this paper, we consider 1D random Clifford encoding circuits and demonstrate that, when the circuits have a logarithmic depth $d = O(\log n)$, the generated codes can be efficiently decoded and can achieve a rate close to the hashing bound for depolarizing noise of various strengths. Our numerical thresholds are plotted alongside various analytical bounds in Fig. 1. We obtain our results by developing a tensor-network maximum-likelihood decoder for stochastic Pauli noise that has poly(n) running time when $d = O(\log n)$. The combination of a high threshold against stochastic noise, nonzero rate, the practicality of low depth in 1D, and efficient decoding shows that such codes are promising candidates for quantum memories in future implementations of fault-tolerant quantum computers.

We remark that the performance of these codes against Pauli noise cannot be derived in a simple way from the results about erasure noise in previous works [30]. While there is a simple relationship between the distance w and the maximum error weight that can be corrected for erasure and Pauli errors with optimal decoding (namely the code can correct any Pauli error up to weight $w/2 - 1$ or any erasure error up to weight $w - 1$), the distance alone does not determine the performance of the code.

For instance, a quantum error-correcting code may suppress errors even when the typical errors have weight much larger than the distance [a simple example of this would be the surface code, where typical errors have weight $O(w^2)$]. This is because a weight $O(w)$ physical error will only cause a logical error if the physical error locations are in a particular configuration, which is unlikely in the case of IID noise [32]. Another reason why the distance alone does not determine performance is that it is not even guaranteed that a quantum error-correcting code can correct up to weight w Pauli errors when heuristic/suboptimal decoding is employed (as is usually the case for Pauli noise).

While most of the analysis in Gullans *et al.* [30] is restricted to the erasure noise model, the paper also contains an analysis of the so-called "block model." This approach can be used to determine Pauli noise thresholds for random codes similar to the ones we have considered in this work. We provide a comparison of the block model to the random circuit encoding of this work to clarify a regime where our random circuit encoding has the advantage over the block model.

This paper is structured as follows. In Sec. I we define the encoding circuits and codes that we focus on in this work. In Sec. II we present the TN decoding algorithm that we use for these codes. In Sec. III we present the results of numerical simulations of these codes using the TN decoder. In Sec. IV we describe the block model of Ref. [30], and we compare it to the codes we study in this work. Finally, in Sec. V we summarize our results and propose directions for future research. Appendixes provide supplementary numerical results related to thresholds, locality properties of the code, and additional comparison between our code construction and the block model.

I. LOW-DEPTH RANDOM ENCODING CIRCUITS

Here we briefly outline the definition of codes based on low-depth Clifford circuits. We start with a trivial quantum code with one-qubit logical operators and checks. We partition the n physical qubits into k logical qubits and $n - k$ stabilizer qubits such that the logical qubits are evenly spaced among the physical qubits. For each stabilizer qubit, indexed by $i \in \{1, \dots, n - k\}$, we randomly associate a nontrivial single-qubit Pauli check operator $g_i \in \{X, Y, Z\}$. The stabilizer of the code is the group $\mathcal{G}$ generated by the checks. The check operators trivially commute, and the code space is defined as the $+1$ eigenspace of all such check operators (or all elements of the stabilizer). This implies that the initial code space is a product state on the stabilizer qubits. The logical qubits, however, are not fixed by the checks, and for every logical qubit, indexed by $j \in \{1, \dots, k\}$, we associate a pair of distinct anticommuting single-qubit Pauli operators l_j^x, l_j^z, which we regard as the logical X and logical Z operators for that qubit.

Given a Clifford circuit U, we can produce a new stabilizer code by transforming the checks and logical operators as $g_i \mapsto U g_i U^\dagger$, $l_j^x \mapsto U l_j^x U^\dagger$, and $l_j^z \mapsto U l_j^z U^\dagger$. We assume U to be noiseless. The circuit U is an encoding circuit that maps unencoded logical qubits to encoded ones. The specific Clifford circuits we consider are low-depth circuits in 1D, where two-qubit iSWAP gates are applied in parallel between neighboring pairs of qubits in an alternating brickwork pattern [see Fig. 1(a) of Ref. [33]]. After each round of two-qubit gates, a uniformly random single-qubit Clifford gate is applied to every physical qubit. The depth d of the circuit is taken to be the number of two-qubit gate layers. These locality constraints imply the weight of each check is at most $2d$.

For the TN decoder, which we define below, it is useful to consider open boundary conditions. To minimize the boundary effect, for a code with a given n and r, we add an additional $4d - 1/r + 1$ stabilizer qubits to make sure all logical qubits are at least $2d$ physical qubits away from the boundary before applying the encoding circuit. This changes the rate of the code; however, given that we restrict to $d = O(\log n)$, this does not affect the asymptotic rate as $n \to \infty$. In a slight abuse of notation, we use r to refer to the rate of the code before the boundary qubits are added, and we define $n_{\text{phys}} := n + 4d - 1/r + 1$ to be the total number of physical qubits (including boundary qubits) of the code.

II. TENSOR-NETWORK DECODING

To assess the performance of these codes, we consider the following scenario. First, the logical information is encoded into an error-correcting code using the encoding circuit U as described above. Next, every physical qubit suffers noise, which we assume is depolarizing noise. Finally, all of the checks are measured, and decoding is performed. We assume the measurements are noiseless. Decoding is a classical computation that takes the check measurement outcomes, called the syndrome, as input and outputs a correction to restore the encoded data. For stochastic Pauli noise, this decoding task (a classical computational problem) is hard in general [34]; however, we show that efficient near-optimal decoding is possible for 1D random Clifford codes of depth $d = O(\log n)$. Note that, while this scenario is not fully realistic due to the assumptions of noiseless encoding circuits and measurements, these assumptions are conventionally used to quantify code performance in a code capacity setting [35]. Here we briefly outline how the decoding problem can be cast as a tensor network (TN) contraction. Say an n-qubit Pauli error e_p occurs on a state in the code space and all of the checks are measured, yielding the syndrome outcomes $s = s_1, s_2, \ldots, s_{n-k}$, where $s_i \in \{-1, 1\}$ is the outcome of the measuring check g_i. Let f be any product of Pauli operators that is consistent with that syndrome, in that it anticommutes with the checks that returned a -1 outcome and commutes with the other checks. The physical error e_p is one such error, but it is not known to the experimenter. One can always efficiently find an operator consistent with a syndrome by performing row-reduction on the check matrix, the rows of which are binary vectors representing the check operators.

The operator f applied to the code will correct e_p if $e_p \in f\mathcal{G}$. If not, then $fe_p = L$ for some nontrivial logical operator L, and fLg for any $g \in \mathcal{G}$ will correct the error. Maximum-likelihood decoding finds the correction that is most likely to correct the error. To do this, we determine the L for which $p(fL\mathcal{G})$ is maximized, i.e., the coset $fL\mathcal{G}$ that e_p most likely belongs to. For any L, we henceforth combine the f and L into a single Pauli error f_L.

The probability of an error belonging to the coset $f_L\mathcal{G}$ is simply the sum of the probabilities of every error in that coset, i.e.,

$$p(f_L\mathcal{G}) = \sum_{e\in\mathcal{G}} p(f_L e). \tag{1}$$

For independent Pauli noise, each summand $p(f_L e)$ is a product of Pauli error probabilities. It is inefficient to compute Eq. (1) directly by evaluating the sum since the number of terms in the summation is $|\mathcal{G}| = 2^{n-k}$. In fact, we do not expect an efficient algorithm to exist for computing coset probabilities of codes in general, due to the #P-hardness of the problem [34].

Fortunately, for codes with local checks, there are examples where coset probabilities can be computed efficiently using TN methods. Here we present two such methods for evaluating coset probabilities that are efficient and nearly optimal when restricted to 1D codes with log-depth encoding circuits. The key is to construct a tensor-network description of Eq. (1) that can be efficiently contracted.

To obtain a tensor network description of Eq. (1), we use the fact that every element of $\mathcal{G}$ is a product of generators, i.e., is of the form $e(\sigma) = \prod_{i=1}^{n-k} g_i^{\sigma_i}$, where g_i is the ith check, and $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_{n-k}$, $\sigma_i \in \{0, 1\}$ for each i represents a particular check configuration.

For independent Pauli noise, we can write the probability of an error $f_L e(\sigma)$ as a product of single-qubit Pauli error probabilities $p(f_L e(\sigma)) = \prod_{j=1}^{n} A_\sigma^{(j)}(L)$, where $A_\sigma^{(j)}(L) \in \{p_I^{(j)}, p_X^{(j)}, p_Y^{(j)}, p_Z^{(j)}\}$ for each σ and where $p_Q^{(j)}$ is the probability of Pauli error Q occurring on qubit j. Note that, if the noise is identical for each qubit j, $p_P^{(j)}$ $(P = I, X, Y, Z)$ does not depend on j. To simplify notation, we will henceforth drop the explicit dependence of $A_\sigma^{(j)}(L)$ on L.

While $A_\sigma^{(j)}$ for a particular site j depends on the check configuration σ, it is clear that it only depends on the bits σ_i for which g_i acts nontrivially on qubit j. Thus, each term $A_\sigma^{(j)}$ can be written as a tensor of rank r_j, where r_j corresponds to the number of generators that act nontrivially on j. For a general stabilizer code, we replace $A_\sigma^{(j)}$ with $A_{\sigma(j)}^{(j)}$, where $\sigma(j)$ is the list of all check bits σ_i for which g_i acts nontrivially on j.

To obtain the coset probability, we sum over all indices

$$p(f_L\mathcal{G}) = \sum_{\sigma} \prod_{j=1}^{n} A_{\sigma(j)}^{(j)}. \tag{2}$$

Note that the expression on the right-hand side has a similar form to a TN contraction. Each $A_{\sigma(j)}^{(j)}$ is a tensor of rank r_j, and the sum of the product structure is nearly identical to tensor contraction. One small difference is that each index σ_i can appear in more than two tensors (they essentially correspond to hyperedges of the TN, compared to usual graph edges, that only connect two nodes/tensors). We can convert this into a typical TN in two ways. One of them is based on the TN description of coset probabilities for the surface code in Ref. [36], which we describe in Sec. II A.

The results presented in this paper, however, have been computed using an alternative TN description of the coset probabilities, which we explain in detail in Sec. II B. We emphasize that these two tensor network descriptions both evaluate to Eq. (1); however, due to their different structure, they require different contraction methods and therefore will take different amounts of time to evaluate (despite both being polynomial time). We have used the version in Sec. II B in our simulations since our implementation of it was substantially faster.

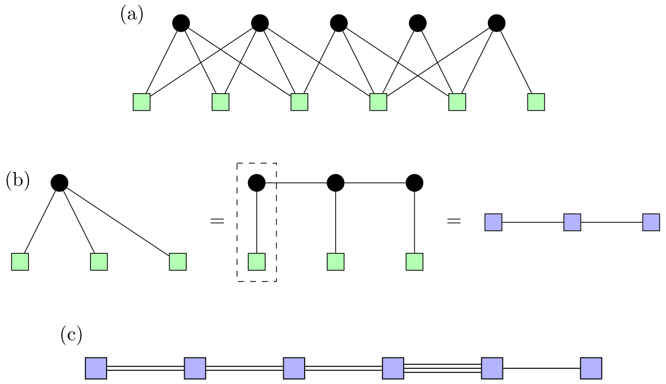

FIG. 2. (a) Coset probabilities represented as a TN following the construction in Ref. [36]. Each black circle node corresponds to a check, and each green square node corresponds to a qubit. An edge is drawn between a check node and a qubit node if and only if the check acts nontrivially on that qubit. The check nodes correspond to δ tensors, and the green nodes correspond to $A_{\sigma(j)}$ tensors (defined in the text). (b) δ tensor can be split and combined with the qubit tensors, resulting in the blue square tensors. (c) This is done for every δ tensor in (a), resulting in a 1D TN. For 1D codes with depth $O(d)$ encoding circuits, the maximum number of edges connecting a pair of neighboring tensors is $O(d)$, resulting in a maximum matrix size of $2^{O(d)} \times 2^{O(d)}$.

A. TN Description of coset probabilities, based on Ref. [36]

One way to turn the summation in Eq. (2) into a TN, as done in Ref. [36], involves adding a δ tensor for each check, with only two nonzero entries specified by $\delta_{\sigma_1,\sigma_2,\ldots,\sigma_n} = 1$ if $\sigma_1 = \sigma_2 = \cdots = \sigma_n = 0$ or 1. The TN is defined by connecting indices of A tensors to the appropriate check tensors δ.

The TN then has the same form as the Tanner graph of the code, as illustrated in Fig. 2. This is a bipartite graph with two types of nodes, which we refer to as qubit nodes and check nodes. Each qubit node corresponds to a physical qubit, and each check node corresponds to a code check. An edge is added between a qubit node and a check node if and only if the check acts nontrivially on the corresponding qubit. This graph is mapped to a TN by placing a δ tensor at every check node and an A tensor at every qubit node.

This TN description has proven useful for the surface code and other planar codes [37] since it can be contracted efficiently using established TN methods. However, for families of codes whose check weight grows with n, like the random codes studied in this paper, one encounters the problem that the size of the tensors grows exponentially with the weight of the checks. Furthermore, the resulting TN is not planar and so the methods of Ref. [37] cannot be applied directly.

Nevertheless, for 1D Clifford encoding circuits, the TN can be converted into a 1D TN as shown in Fig. 2. The 1D TN consists of a product of (sparse) matrices of size $2^{O(d)} \times 2^{O(d)}$. When $d = O(\log n)$, this TN can be contracted exactly in polynomial time in n.

We have implemented this decoder and confirmed that it produces the correct coset probabilities. We did not use this TN description to obtain the results in this paper, since our implementation of it was slower than the method described in the following section. We note, however, that this description could prove useful if optimized, e.g., by exploiting the sparse structure of the matrices. For the remainder of this section, we focus on the following alternative TN of coset probabilities.

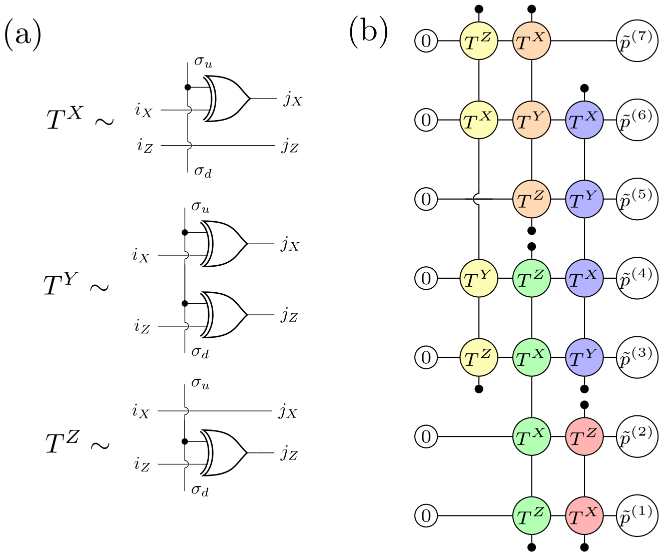

FIG. 3. (a) Tensors defined in terms of logic circuits. The entries of these tensors are 1 for any assignment of bits to the tensor indices that are valid with respect to the circuit, and 0 otherwise. In particular, it is 1 only if $\sigma_u = \sigma_d$ for each tensor. The logic gate appearing in these circuits is the XOR gate. In computing $p(f_L e)$, where $e \in \mathcal{G}$, the indices σ_u and σ_d specify if a stabilizer generator is contained in e or not, and (i_X, i_Z) and (j_X, j_Z) are for keeping track of changes of Pauli operators if the generator is applied. (b) The coset probabilities for maximum-likelihood decoding for any stabilizer code can be expressed as the contraction of a two-dimensional TN. Each horizontal row corresponds to a physical qubit of the code, so the number of rows is always n_{phys}. The network is constructed from generators of $\mathcal{G}$ (checks). In this illustration, nodes corresponding to different generators are distinguished by their color. As illustrated, the generators are sorted into columns and are arranged such that no two generators overlap in a given column. Nontrivial Pauli operators in a generator are replaced with the corresponding tensors in (a), and all tensors in a generator are connected by a vertical wire. This implies that all σ_u and σ_d in the tensors corresponding to a given check must coincide to contribute a nonzero term to the contraction. The tensor $\tilde{p}^{(j)}$ is a vector of the four Pauli error probabilities on site j which are permuted according to the action of f_L on site j. The 0-tensors on the left fix the left indices to 0, and the small black tensors have entries all equal to 1. By horizontally contracting the tensors in a given row with σ indices fixed, either p_I, p_X, p_Y or p_Z on the jth qubit is obtained depending on what Pauli operators act on the qubit in $f_L e$. Contracting the network corresponds to summation over all σ indexes, and thereby all $e \in \mathcal{G}$, which evaluates to $p(f_L \mathcal{G})$ as in Eq. (1).

B. Alternative TN description of coset probabilities

Here we describe an alternative way to efficiently represent the coset probabilities $p(f_L \mathcal{G})$ for any stabilizer code as a two-dimensional network of small tensors. We have used this method to produce the results presented in this paper. The network is illustrated in Fig. 3.

Most of the tensors in the network are from the set $\{T^X, T^Y, T^Z\}$, which are defined in Fig. 3(a) in terms of classical logical circuits. The tensors have an entry 1 for each index assignment corresponding to a valid execution of the circuit, and the remaining entries are zero. The tensors are precisely defined as follows: for $P = X, Y, Z$, $T^P_{i_X,i_Z,j_X,j_Z,\sigma_u,\sigma_d} = 0$ if $\sigma_u \neq \sigma_d$,

$$T^P_{i_X,i_Z,j_X,j_Z,0,0} = \begin{cases} 1 & \text{if } (i_X, i_Z) = (j_X, j_Z), \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

and

$$T^X_{i_X,i_Z,j_X,j_Z,1,1} = \begin{cases} 1 & \text{if } (i_X, i_Z) = (j_X \oplus 1, j_Z), \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

$$T^Y_{i_X,i_Z,j_X,j_Z,1,1} = \begin{cases} 1 & \text{if } (i_X, i_Z) = (j_X \oplus 1, j_Z \oplus 1), \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

$$T^Z_{i_X,i_Z,j_X,j_Z,1,1} = \begin{cases} 1 & \text{if } (i_X, i_Z) = (j_X, j_Z \oplus 1), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

As we will see below, in the computation of the probability $p(f_L e)$ $(e \in \mathcal{G})$, the indexes σ_u and σ_d are used to characterize what generators $g_i \in \mathcal{G}$ are contained in e, and the indexes i_X, i_Z, j_X, j_Z are used for recording the changes of the Pauli operators when each such g_i is applied. Following the graphical representation of the tensor as shown in Fig. 3(a), in the following, we may refer to (i_X, i_Z), (j_X, j_Z), σ_u, and σ_d at the left, right, top, and bottom indexes of the tensor, respectively.

A tensor $\tilde{p}^{(j)}$, which is dependent on f_L (although not explicitly in our notation), is associated with the Pauli error probabilities p_I, p_X, p_Y, p_Z occurring at the jth qubit. This tensor has two left-pointing binary indexes (i_X, i_Z) and has entries from $\{p_I, p_X, p_Y, p_Z\}$. In the case of $f_L = I$, the entries of the tensors $\tilde{p}^{(j)}_{(i_X,i_Z)}$ equal p_I, p_X, p_Y, p_Z for $(i_X, i_Z) = (0, 0), (1, 0), (1, 1), (0, 1)$, respectively. For nontrivial f_L, one simply flips input bits of $\tilde{p}^{(j)}$ according to the action of f_L on site j, so if $f_L^{(j)} = X$, one flips only the i_X bit in the definition above, and if $f_L^{(j)} = Z$, one flips i_Z and for $f_L^{(j)} = Y$ one flips both bits.

We now explain how to determine the layout of the 2D network. The number of rows in the network is always n_{phys}, but the number of columns depends on the details of the checks, which we will explain later. Given a single check, we construct part of the network corresponding to that check as follows. When the check acts as a nonidentity Pauli operator P on the jth qubit, we place T^P tensor in the jth row. We do this for each qubit that the check acts nontrivially on and connect the $\sigma_{u/d}$ indices of all the T^P tensors by a vertical wire. The σ_u of the top tensor and σ_d of the bottom tensor are each connected to a single index δ tensor, which has two entries both equal to 1 and is indicated by a small black circle in Fig. 3(b). We call this connected set of tensors a check subnetwork.

The set of all check subnetworks, where each subnetwork corresponds to a given check, is then sorted into columns so that no pair of subnetworks in a given column overlap on a row. The right index of every T^P tensor is connected to the left index of the tensor on its right. The left index of the leftmost tensor T^P in each row is fixed to 0. The right index of the rightmost tensor T^P in the jth row is connected to the index of $\tilde{p}^{(j)}$. See Fig. 3(b) for a specific example of the TN.

The number of columns depends on the choice of checks and can be $n - k$ when there is a qubit shared by all the checks. However, in the case of 1D encoding circuits of depth d, the number of columns can be $O(d)$ as the checks can only act nontrivially on at most $2d$ neighboring qubits.

In the 2D TN, each horizontal row is associated with a single physical qubit, and every horizontal wire actually contains two single-bit wires, which we can represent as a single tensor edge of dimension 4. In the jth row, the i_X bit corresponds to an X error on the jth qubit, and the other i_Z corresponds to a Z error on the jth qubit. By contracting with the probability tensor, a probability factor of $p_I^{(j)}$, $p_X^{(j)}$, $p_Y^{(j)}$, or $p_Z^{(j)}$, which does not depend on j if the noise is identical for all qubits, will appear, depending on bit values of the wire. Note that the probability tensor is permuted according to f_L, as we have explained.

In contracting the TN, it is important to notice that the set of T^P tensors in the network constitutes a large logic circuit and that only valid computations of the classical logic circuit will be summed over (invalid computations evaluate to zero). In valid computations of the classical circuit, the bits $\sigma_{u/d}$ along all vertical wires in the tensors associated with single check qubits must all be 0 or all be 1. If $\sigma_{u/d}^{(i)} = 1$ for a given check i, then the bits on the horizontal wires matching that check will be flipped; otherwise, they will not be affected. Fixing the vertical check indices $\sigma_{u/d}^{(1)}, \sigma_{u/d}^{(2)}, \ldots, \sigma_{u/d}^{(n)}$ and summing over the remaining indices, we can pull out a product of Pauli probabilities from the $\tilde{p}$ tensors, which is equal to $p(f_L e(\sigma))$, with $\sigma = \sigma_{u/d}^{(1)}, \sigma_{u/d}^{(2)}, \ldots, \sigma_{u/d}^{(n)}$. Finally, in contracting the TN by summing over the check indices, we sum over all possible configurations σ of the check bits and obtain the coset probability $p(f_L\mathcal{G}) = \sum_\sigma p(f_L e(\sigma))$.

C. Contracting the network

The TN described in Sec. II B can be contracted in various ways. The task is, essentially, to contract an $L \times W$ sized square-lattice TN, where L and W are the length and width, respectively, of the network. To perform the contraction exactly, we first contract the first row into a single tensor with $O(W)$ indices. We then contract the remaining tensors one by one with this tensor, starting with the first tensor in the second row, then moving down the network along rows and then down columns until all tensors are contracted. Assuming $L \geqslant W$, the maximum memory cost of this procedure is $O(2^W)$, and the time cost is $O(WL2^W)$. For the low-depth random circuits we consider, $L = n_{\text{phys}}$ and $W = O(d) = O(\log n_{\text{phys}})$, and so both the time cost and memory cost are polynomial in the block size. Note that this is in contrast to the problem of contracting an $n \times n$ square lattice TN, which is known to be #P-complete [38].

One could try to improve this scaling further by, e.g., using approximate contraction strategies. The approximate boundary matrix product state (MPS) method, described in Ref. [39] and used in other decoders [36,37,40,41], would reduce the time and memory costs to polynomial in W if a truncated bond dimension χ is kept constant in N. We have tested this method, but it appears that the quality of the approximation varies considerably among the codes sampled. A direction of future research could be to find methods to contract the network more efficiently. However, for this work, we have found that the exact contraction method is sufficiently fast for our purposes.

D. Decoding logical qubits independently by marginalization

The coset probability $p(f_L\mathcal{G})$ can be computed for any L using this method. One problem when encoding a large number of qubits k is that the number of inequivalent logical operators and cosets grows as 4^k. Therefore, finding the most likely coset by computing the probability of every coset takes exponential time in the number of encoded qubits. We overcome this issue by decoding each logical qubit independently.

To decode a single logical qubit j, we calculate the probabilities of the cosets for $f_L \in \{f, fUl_j^xU^\dagger, fUl_j^zU^\dagger, fUl_j^xl_j^zU^\dagger\}$ and marginalize over the other logical qubits, where l_j^p are the unencoded logical Pauli operators for the jth logical qubit, and U is the encoding circuit. This marginalization is achieved simply by adding the logical generators $Ul_i^xU^\dagger, Ul_i^zU^\dagger$ for all i different from j to the list of checks when we construct the TN. In the case of the 1D encoding circuit with depth d, adding the logical operators to the TN adds an additional $O(rd)$ columns to the network and therefore does not change the scaling of the width of the network with d. However, the cost of marginalization on exact TN contraction does increase with r, which explains why our simulations can only deal with $d = 7$ with $r \geqslant 1/3$, while $d = 8$ for $r \leqslant 1/5$.

This approach returns the optimal correction for any given logical qubit, but may not be globally optimal due to possible correlations between errors on logical qubits. However, we do not expect this approximation to affect performance. The reason for this is that when error correction is working, the probability of the most likely coset should be substantially higher than the sum of the probabilities of all other cosets. If not, then even with an optimal correction, there would still be a large probability of logical error. When one coset has a substantially larger probability than all other cosets combined, the optimal correction for any given logical qubit obtained by marginalization will agree with the global optimum.

A TN must be contracted for each coset of each logical qubit, so $4k$ contractions must be evaluated in total. Fortunately, most of the contraction (i.e., the contraction of tensors on qubits on which the logical does not act) can be reused, and thus decoding every qubit requires only a small additional cost compared to decoding a single logical qubit.

III. NUMERICAL RESULTS

We have performed simulations using the TN decoder described above to study the properties of codes defined by random Clifford encoding circuits in 1D. In each run of the simulation, we randomly generate a code using a Clifford circuit of depth $d = O(\log n)$ as described above and sample a Pauli error e_p according to the depolarizing noise model, which gives rise to a syndrome s. The decoder calculates a correction f_L, using s as input.

We say that logical qubit j fails when f_Le_p anticommutes with at least one of the logical generators $Ul_j^xU^\dagger$ or $Ul_j^zU^\dagger$, which can be easily checked. At least 2×10^5 runs of the simulation are performed for each data point. Taking the average over the stochastic Pauli noise as well as over the random Clifford encodings, we can estimate the probability of any given logical qubit failing.

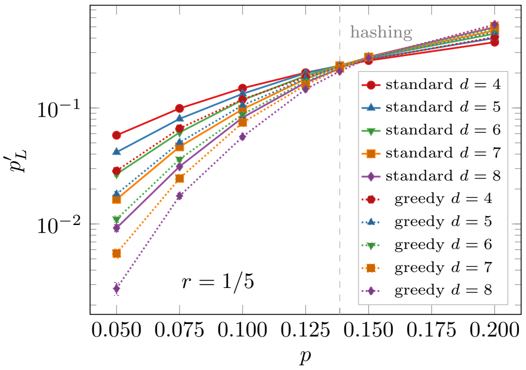

FIG. 4. Bulk logical error probability p'_L vs physical error probability p for $r = 1/5$ using a fixed system size of $n = 50$, excluding $O(d)$ added boundary qubits using both standard and greedy random encoding circuits. A clear crossing point can be observed, which is very close to the threshold error probability implied by the hashing bound $p = 0.139$ indicated by the dashed gray line. Computed thresholds and threshold plots for other rates are included in Appendix B.

First, although there is a clear boundary effect, we observe that the logical failure probability for qubits sufficiently far from the boundary is uniform across logical qubits and independent of system size if d is kept constant. We let p'_L denote the failure probability of a logical qubit in the bulk region. We also find that, unlike the fully random stabilizer codes, the spatial correlations of logical errors are short-ranged, as observed in Ref. [30] for erasure noise. We present numerical evidence for these properties in Appendix A.

We have plotted p'_L as a function of the physical error probability for a variety of depths d and $r = 1/5$ in Fig. 4. The crossing point for curves of different d indicates a threshold in the code, below which p'_L decays exponentially in d. Evidence of this exponential decay, plots at other rates, as well as a list of computed thresholds are included in Appendix B. The numerically obtained thresholds are plotted alongside the hashing bound in Fig. 1.

The performance of the code can be improved by slightly modifying the random Clifford circuits. As one instance, we propose a random "greedy" code by choosing single-qubit gates in the random circuit to maximize the weight of checks and logical generators in each layer, rather than uniformly at random. See Appendix C for the details. The results of the greedy construction are also provided in Fig. 4 alongside the standard construction. Despite differences in logical error rates, the threshold crossing points are very similar, and the greedy construction appears to behave like the standard construction except with a greater effective depth.

For both constructions, these plots show that the thresholds are very close to the hashing bound for $r \lesssim 1/3$. Hence these codes achieve the same threshold as a fully random code for these rates, despite being much more local and restricted. At higher rates, e.g., $r = 1/2$, a threshold is harder to discern from the data; however, we conjecture that using larger values

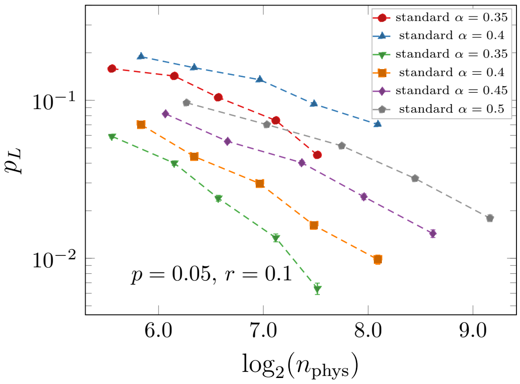

FIG. 5. Probability of at least one logical qubit failing vs $n_{\rm phys}$ when $d = \alpha^{-1} \log k$, for various α. The error rate decays to zero with $n_{\rm phys}$, and by varying α we can increase the number of encoded qubits at the cost of a slower rate of decay.

of d than is currently accessible with our numerical method would produce a threshold estimate close to the hashing bound, as with the lower rates.

Note that the observed exponential rate of decay in d of p'_L below the threshold for $r \lesssim 1/3$ implies a polynomial decay in p'_L in n when $d = O(\log n) = O(\log k)$. If we define α, δ, and D such that $\log k = \alpha d$, and $p'_L = D \exp(-\delta d)$, then the probability of an error occurring on at least one logical qubit, which we refer to as p_L, satisfies $p_L = D k^{1-\delta/\alpha} + O(p'^2_L)$. Therefore, p_L can be made to decrease polynomially fast by setting $\alpha < \delta$. Note that log, without a subscript, always refers to the natural logarithm, and although not explicitly indicated in our notation, in expressions where two integer values are related (e.g., k and d in $\log k = \alpha d$), we always assume that the dependent variable is rounded to the nearest integer. In Appendix B, we have listed the exponents δ for various values of r and p, from which one can determine the values of α that result in asymptotically zero logical error probability. Note that $1 - p_L$ is equal to the entanglement fidelity of the k-qubit logical channel (which is proportional to the average channel fidelity [42]).

In Fig. 5, we plot p_L as a function of $n_{\rm phys}$ with fixed rate $r = 0.1$ and $p = 0.05$ and various α. As can be seen, by varying α, we observe a tradeoff between the rate of decay of the total logical error rate and the total number of encoded qubits $k = rn$. For practical error correction, it would likely be useful to tune the value of α as well as the rate r, according to the target number and error probability of logical qubits.

We remark that only a polynomial decay in logical error probability is possible for constant rate r when $d = O(\log n)$. While an exponential decay in logical error probability is desirable, polynomial decay is sufficient for the use of implementing quantum algorithms.

IV. BLOCK MODEL

In the previous section, we saw that 1D random codes defined in Sec. I achieve a threshold close to the hashing bound. To shed some light on this, we now consider a more restricted and simpler "block model" with logarithmic block sizes, which was described previously in Ref. [30]. Compared to the random circuits we have described above, the threshold of the block model is easy to determine analytically.

In the block model, the 1D chain of n physical qubits is partitioned into noninteracting blocks of size $m = \beta \log_2 n$. Note that the block model in this paper always refers to the one with logarithmic block sizes. We then use encoding circuits which have the restriction that no two-qubit gates couple qubits from different blocks. This model therefore describes a tensor-product of independent codes, each with $O(\log n)$ physical qubits.

Each block can be made into a random stabilizer code with a 1D $d = O(m) = O(\log n)$ random Clifford encoding circuit. For a random Clifford encoding, under an IID Pauli noise model, in the limit of $n \to \infty$, the logical error probability of each individual block will tend to 0 exponentially fast in the block size m below the hashing bound. Hence the threshold of the whole code composed of $O(n/\log n)$ independent blocks will also equal the hashing bound provided that β is sufficiently large. See below for an asymptotic evaluation of such β.

Polynomial time maximum-likelihood decoding is also possible in this block model by calculating the coset probabilities in Eq. (1) by brute force summation, which takes exponential time in the block size or polynomial time in n when the block size is $O(\log n)$.

While the block model has many of the appealing properties of the 1D circuits we have studied above, the lack of couplings between blocks appears to compromise its error-correcting performance. We have performed numerical simulations of the block model using the TN decoder described in Sec. II (which we expect to be near optimal).

In Fig. 6, we present results comparing the logical error rates for a given encoding depth of the block model to the standard and low-depth random circuit encoding studied in previous sections. In Fig. 6(a), we choose $r = 1/3$ and $p = 0.025$, which is somewhat below the hashing bound, while in Fig. 6(b), we choose $r = 1/2$ and $p = 0.07$, which is only slightly below the hashing bound.

In the former case, we see that the logical error rate for a given encoding circuit depth d is substantially higher for the block model than for the low-depth random circuit encoding (for both the standard and greedy code constructions). In Appendix D, we present data that show the block model also has worse performance in the case when the logical error rate is plotted against the average check weight rather than encoding circuit depth d. Thus, while the threshold is easy to determine with the block model, we have demonstrated that in certain regimes the low-depth random circuit encoding offers superior performance without being more complex to implement in practice.

Although we do not present the numerical results here, essentially the same trend is also observed when the logical error rate is taken to be the probability of at least one qubit failing on a region of size proportional to d (rather than the average failure rates of each qubit).

The data in Fig. 6(b) close to the threshold do not show a clear advantage or disadvantage of the block model compared to the circuit encodings. However, they highlight that small-size effects are likely the reason why a clear threshold is not observable in our $r = 1/2$ data (see Fig. 9 in Appendix B)

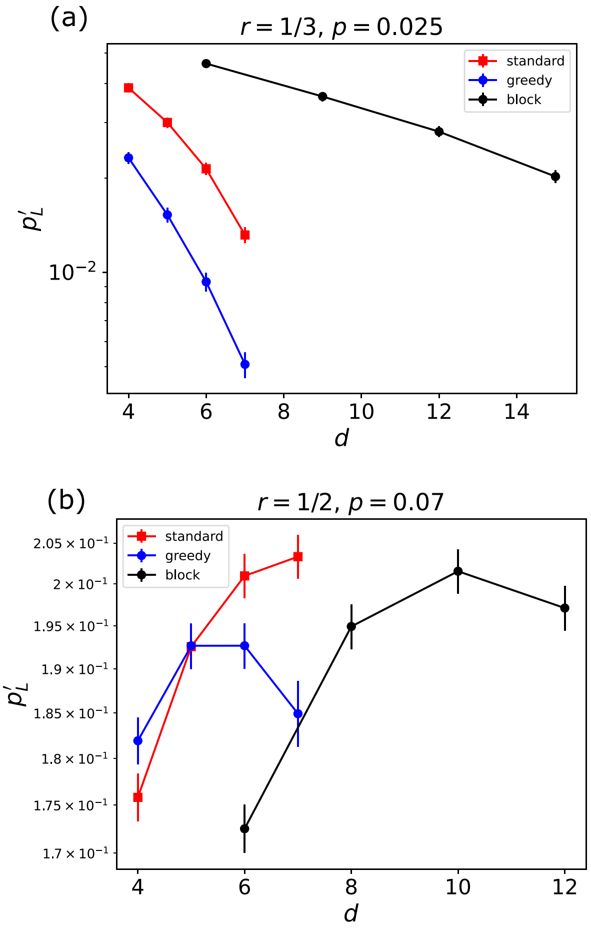

FIG. 6. Comparison of logical error rates of 1D random codes to the block model described in Sec. IV, in which the n qubits are separated into independent blocks of size $O(\log n)$ for an error rate (a) somewhat below the hashing bound, and (b) slightly above the hashing bound. The quantity p'_L on the y-axis is the average logical qubit error rate, which for the greedy and standard code constructions is taken over logical qubits sufficiently far from the boundary to avoid boundary effects. The quantity d on the x-axis is the encoding depth of the circuit for the standard and greedy constructions and is the block size for the block model (which is a lower bound on the circuit depth required to implement a uniformly random Clifford unitary on a block with using nearest-neighbor gates in 1D). For the greedy and standard code constructions, we have used $n = 54$ and 50 in (a) and (b), respectively.

for the standard and greedy constructions. Neither the block model nor the other circuit encodings exhibit an exponential decay in logical error probability p'_L in this region, despite the fact that asymptotically the block model should have exponential decay for any p below the hashing bound.

The fact that the block model provably achieves the hashing bound suggests that the low-depth random circuit encoding may also achieve the hashing bound. However, we do not currently have rigorous analytical proof of this, and we leave this as an open question for future investigation.

To evaluate β discussed above, finite-size effects in the block model would also be an important factor that we should carefully consider. While finite-size effects exist in the low-depth random circuit encoding as well, those in the block model can be more severe due to the logarithmic block size and may result in worse performance than naively expected from the asymptotic achievable bound, i.e., the hashing bound.

Let q_L be the probability of an error occurring on at least one logical qubit in the block model against the IID Pauli noise. When $m = \beta \log_2 n$, we can show that $q_L \leqslant \frac{1}{\beta \log_2 n} n^{1-\frac{\beta}{4}R}$ to the leading order of n, where $R = 1 - H(\vec{p}) - r$. This reveals the asymptotic behavior of q_L. If $R > 0$, i.e., if $(r, \vec{p})$ is below the hashing bound, and $\beta \geqslant 4/R$, q_L decreases polynomially in n for sufficiently large n. Note that β diverges when R goes to zero, possibly indicating that a large block size is needed to asymptotically achieve the hashing bound.

We can similarly derive finite-size corrections to q_L. To the second-order of n, we obtain

$$q_L \lesssim \frac{1}{\beta \log_2 n} n^{1-\frac{\beta}{4}R+\frac{\gamma(\vec{p})}{4}\sqrt{\frac{\beta}{\log_2 n}}}, \tag{7}$$

with a coefficient $\gamma(\vec{p}) > 0$. The last term in the exponent is the second-order correction and quantifies the finite-size effect in the achievable bound. As the term scales as $(\log_2 n)^{-1}$, the finite-size effects can remain significant in the block model even for fairly large n.

From Eq. (7), we can estimate the block sizes required for the finite-size effect to be negligible, which turn out to be significant especially in the vicinity of the hashing bound. To demonstrate this, we consider the IID Pauli noise with $\vec{p} = (1 - p, p/3, p/3, p/3)$. For $(r, p) = (1/3, 0.025)$, the finite-size effect in Eq. (7) is negligible when the block size is moderately large around a few tens to a hundred, while much larger block sizes, such as $m = 10^4$–10^5, are required for $(r, p) = (1/2, 0.07)$ that is slightly below the hashing bound. As $m = \beta \log_2 n$, the system size n should be accordingly huge.

Although these are about the finite-size corrections to the achievable bound, i.e., the hashing bound, they may indicate that the performance of the block model in the intermediate-scale systems would be far worse than naively expected from the asymptotic case. We provide a more in-depth analysis of finite-size corrections in Appendix D.

V. DISCUSSION AND CONCLUSIONS

In this work, we have studied quantum error correcting codes defined by 1D low-depth Clifford encoding circuits. We have shown that for the family of these codes with logarithmic depth $d = O(\log n)$, maximum-likelihood decoding of stochastic Pauli noise can be performed in poly(n) time using TN methods. We have also numerically shown that, for depolarizing noise over a large range of noise strengths, the codes can achieve a rate close to the hashing bound if $d = O(\log n)$. Thus, 1D Clifford encoding circuits with depth $O(\log n)$ can generate quantum error correcting codes that have the same rate as random stabilizer codes and can be efficiently decoded. The high-performance, nonzero rate as well as locality in 1D suggest that such codes could serve as practical quantum memories in future implementations of quantum computers.

These results suggest a number of potential directions for future research. First, it would be interesting to see whether codes defined by low-depth Clifford encoding circuits in two or higher dimensions have advantages over the 1D codes considered here. This was shown to be the case for the erasure noise model when the code is modified with a process called expurgation [30].

Another direction towards practical implementation is to consider the realistic case where syndrome extraction is itself prone to error. In this case, fault-tolerant methods for state preparation and logical gates must be developed. One potential route is via Knill's fault-tolerant error correction gadgets that work for any stabilizer code even with a nonconstant-weight stabilizer like ours [43,44]. Finally, it would be interesting to see whether analytical proofs of the rates of these codes can be made by, for instance, strengthening bounds in Ref. [29].

ACKNOWLEDGMENTS

A.S.D. was supported by JST, PRESTO Grant No. JPMJPR1917, Japan and Grant-in Aid for Transformative Research Areas (A) 21H05183. Y.N. is supported by MEXT-JSPS Grant-in-Aid for Transformative Research Areas (A) Extreme Universe, Grants No. JP21H05182 and No. JP21H05183, and by JSPS KAKENHI Grant No. JP22K03464. S.T. was supported by JST Moonshot R&D Grant No. JPMJMS2061. H.Y. was supported by JST PRESTO Grant No. JPMJPR201A and MEXT Quantum Leap Flagship Program (MEXT QLEAP) JPMXS0118069605, JPMXS0120351339. The numerical computation in this work was carried out at the Yukawa Institute Computer Facility.

APPENDIX A: LOCALITY OF LOGICAL ERRORS

In this Appendix, we discuss the locality properties of the codes defined by 1D low-depth encoding circuits with the TN decoder. In particular, we show how the logical error rate of a bulk qubit tends to a system-size-independent value, and how correlations between logical qubit failures are short-ranged.

In Fig. 7, we show the spatial distribution of logical errors for various system sizes. Although there is a clear boundary effect, we observe that the logical failure probability for qubits sufficiently far from the boundary is uniform across logical qubits and independent of system size if d is kept constant. We let p'_L denote the failure probability of a logical qubit in the bulk region.

We also observe that correlations between failures for log-depth circuits under Pauli noise are short-ranged, as was observed in Ref. [30] for erasure noise. Let $P_{2|1}$ be the probability that qubit 2 fails given that qubit 1 fails, and P_2 the probability that qubit 2 fails. In Fig. 8, we have plotted the difference of these quantities, i.e., to what extent the failures are correlated, as a function of the separation x between qubit 1 and 2, normalized by R and d and averaged over all qubit-2 locations. It can be seen that, with this normalization, the various curves collapse, and correlations have a finite range on the order of rd. This indicates that, unlike fully random codes, these codes retain some aspects of the spatial locality under local Pauli noise.

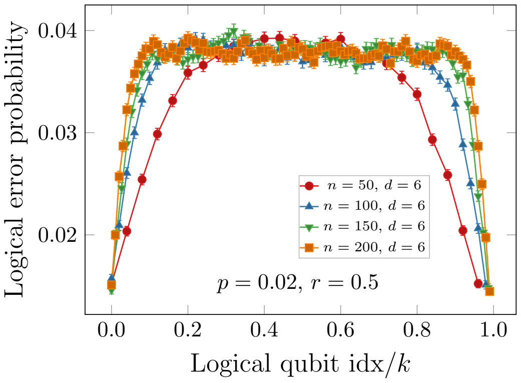

FIG. 7. Error probability of a logical qubit vs a logical qubit index divided by k (corresponding to its relative position on the chain) for $r = 0.5$ with $d = 6$ and $p = 0.02$. The boundary effect can clearly be seen. The error rate stabilizes to a system-size independent value at a certain distance from the boundary.

APPENDIX B: SUPPLEMENTARY NUMERICAL RESULTS

In this Appendix, we provide additional results and details on the numerical methods. The thresholds in this paper are estimated using the critical exponent method similar to that described for the surface code in Ref. [45], which assumes that, near the threshold, the logical failure rate depends only on the rescaled variable $x = (p - p_c)d^{-1/\nu}$, where p_c is the threshold, and ν is some critical exponent. Note for the codes with low-depth encoding circuits that we have replaced the lattice dimension (used for the surface code) with the circuit depth d. By fitting the bulk logical error probability p'_L to a quadratic polynomial in x,

$$p'_L = A + Bx + Cx^2, \tag{B1}$$

we obtain p_c, ν, A, B, and C as fit estimates. The obtained threshold values for different r along with the hashing bound are displayed in Table I.

We also include additional threshold plots for various rates in Fig. 9. As can be seen, a clear crossing point is discernible up to $r = 1/3$.

Finally, in order to illustrate the exponential decay of p'_L as a function of d, we have plotted p'_L versus d on a semilog plot in Fig. 10. A straight-line relationship is observed

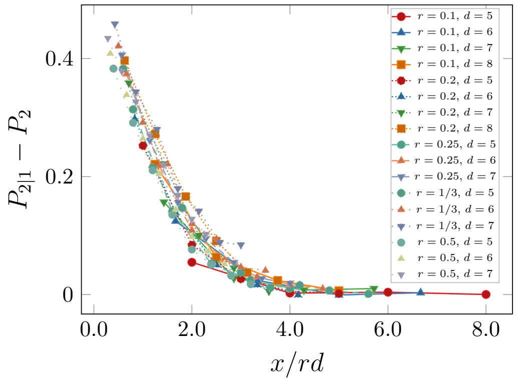

FIG. 8. Two-body correlations between qubit failures vs distance normalized by rd. The curves appear to collapse onto a single line.

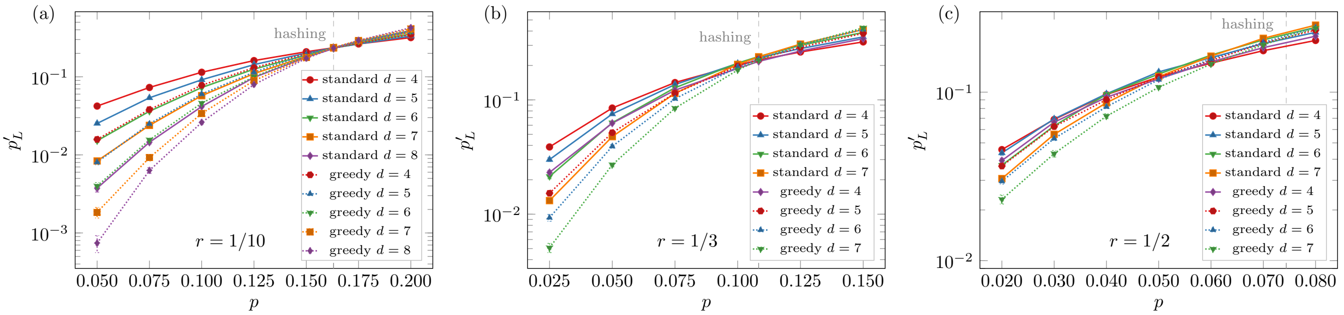

FIG. 9. Bulk logical error probability p'_L vs physical error probability p for various encoding circuit depths d for $r = 1/10$, 1/3, and 1/2 in (a), (b), and (c), respectively. The crossing point of the different curves on each plot corresponds to the threshold. For $r \leqslant 1/3$ the threshold is close to that given by the hashing bound, while a clear crossing point is not discernible for $r = 1/2$ for the values of d simulated. The system sizes used for (a), (b), and (c) are $n = 50$, 54, and 50, respectively.

below the threshold, indicating exponential decay. To be more precise, the exponential decay of p'_L in d is captured by $p'_L = D\exp(-\delta d)$ for some constants D and δ. In Table II, we list the exponents δ computed by fitting p'_L and d to this function for various r and p below the threshold.

APPENDIX C: IMPROVEMENTS TO FULLY RANDOM CLIFFORD ENCODING

Here we describe the "greedy" method to improve the codes produced by random Clifford circuits. This slightly modifies the encoding circuit to avoid low-weight checks and logicals. The specific details of the Clifford circuit (e.g., the choice of the iSWAP as the two-qubit gate) are not expected to make much difference for a high depth, such as $d = O(n)$, where the generated code is expected to be close to a fully random stabilizer code, as exactly shown for a class of Clifford gate set [46]. However, for a low depth, the choice of gate set can make a large difference, as we will observe in the performance improvement in the "greedy" method.

The first simple modification is to ensure that the initial checks are either single-qubit X or Y before applying the first layer of iSWAP gates. This ensures that the weight of all checks is increased to 2 by the iSWAP gates (the weight of a single qubit Z would not be changed). After this step, the distance between the first and last sites on which the check acts nontrivially is guaranteed to be increased maximally by 2 for every layer of iSWAP.

Note that, when a two-qubit gate is applied to a Pauli operator acting on two qubits, the weight of the operator may increase, decrease, or stay the same. Therefore, rather than choosing single-qubit Clifford gates uniformly at random, as in the standard construction, we choose the single-qubit Clifford gate that maximizes the increase in total check weight (sum of the weights of all the checks) when passed through the next iSWAP gate. When multiple gates produce the same

TABLE I. The column p_c (TN) contains the maximum depolarizing error probabilities for error suppression (threshold) we obtain from simulations with codes defined by log-depth 1D random Clifford circuits using the tensor network decoder and fitting our numerical data to Eq. (B1). The parenthesized value indicates the error in the last digit, which is the standard error obtained for the fit estimate. The column p_c (hashing) contains the probabilities given by the hashing bound for this noise model.

r	p_c (TN)	p_c (hashing)
1/10	0.164(2)	0.16305
1/5	0.144(3)	0.13854
1/4	0.125(4)	0.12690
1/3	0.102(3)	0.10835
1/2	0.061(3)	0.07439

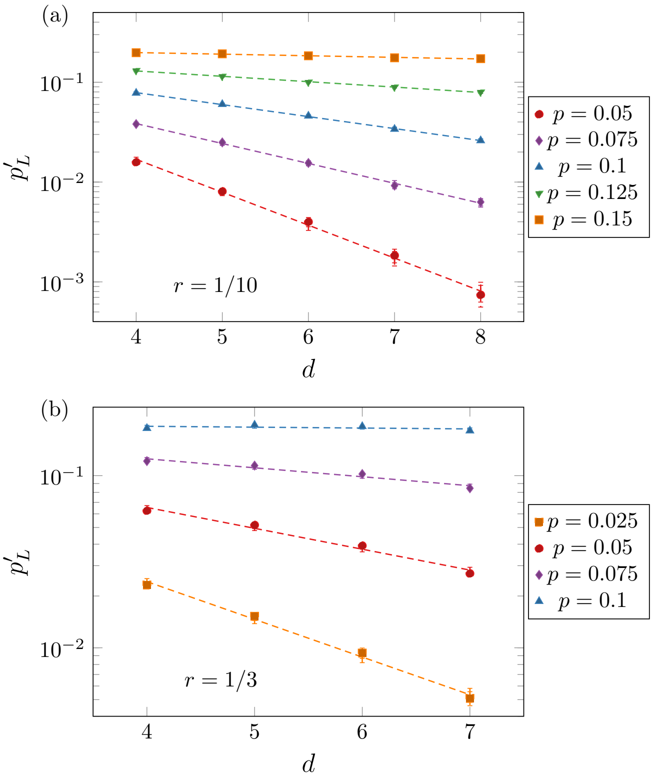

FIG. 10. Bulk logical error probability p'_L vs depth d for $r = 1/10$ and 1/3 in (a) and (b), respectively, for various error probabilities using the greedy code generator. The dashed lines are obtained by linear regression. A straight-line relationship indicates exponential decay in d.

TABLE II. Exponents δ obtained by fitting the numerically computed logical failure probabilities to $p_L' = D\exp(-\delta d)$. The parenthesized value indicates an error in the last digit, which is the standard error obtained for the fit estimate.

r	p	δ
1/10	0.05	−0.72(2)
	0.075	−0.45(1)
	0.1	−0.274(4)
	0.125	−0.125(2)
	0.15	−0.036(2)
1/5	0.05	−0.55(3)
	0.075	−0.33(1)
	0.1	−0.17(1)
	0.125	−0.059(6)
1/4	0.05	−0.44(2)
	0.075	−0.23(1)
	0.1	−0.09(1)
1/3	0.025	−0.48(3)
	0.05	−0.26(3)
	0.075	−0.11(2)

increase in weight, one among these is chosen uniformly at random.

It is harder to avoid low-weight logical operators since any product of logical generators and stabilizers is also a logical operator, and therefore it is harder to check that the overall weight of logical operators increases or decreases with the application of a two-qubit gate. However, as a simple heuristic, we also add the logical generators to the list of checks whose weight is maximized by greedily choosing single-qubit gates. This appears to slightly improve performance.

APPENDIX D: BLOCK MODEL SUPPLEMENTARY INFORMATION

Here we provide some additional details comparing the standard and greedy random circuit constructions to the block model, described in Sec. IV. Figure 6 shows that for values of p somewhat below the hashing bound, the logical error rate of the block model is significantly higher than that of both the standard and greedy circuit encodings for a given encoding depth. It may be tempting to attribute this entirely to the fact that the block model produces codes with lower weight checks on average for a given depth (since checks are prevented from spreading between blocks). However, we see in Fig. 11(a) somewhat below the hashing bound that if we plot the logical error rate against the check weight, the block model still appears to perform worse than the unblocked model. In contrast, we do not observe such a clear advantage of the circuit encodings compared to the block encodings for error rates close to the threshold, as shown in Fig. 11(b).

One potential explanation for the lower performance of the block model is second-order effects in the hashing bound, which we describe in the rest of this Appendix.

In the following analysis of the block model, we leave the block size as m and only assume that m is an increasing function of n. We use the relation $m = \beta \log_2 n$ when it is necessary.

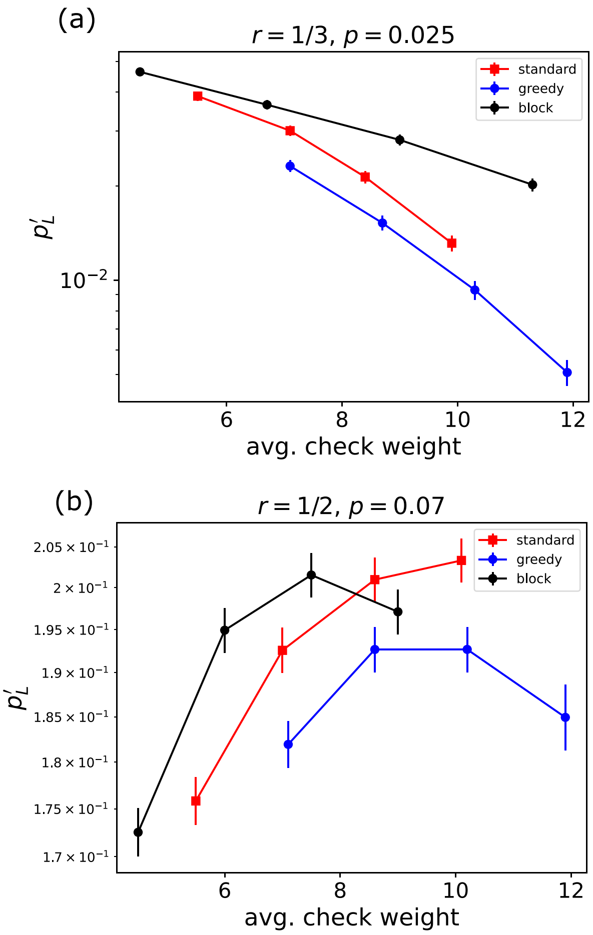

FIG. 11. Logical error rate vs average check weight for block model and circuit encodings. For standard and greedy circuit encodings, the weights averages are taken only for checks sufficiently far from the boundary to avoid finite-size effects.

We first introduce key quantities of the analysis. For two positive-semidefinite operators ρ and σ such that $\operatorname{supp}\rho \subseteq \operatorname{supp}\sigma$, the δ-smooth quantum max-relative entropy, where $0 \leqslant \delta \leqslant 1$, is defined by

$$D^{\delta}_{\max}(\rho\|\sigma) := \min_{\tilde{\rho}\in\mathcal{B}_\delta(\rho)} \inf\{\lambda \in \mathbb{R} : \rho \leqslant 2^{\lambda}\sigma\}, \tag{D1}$$

where $\mathcal{B}_\delta(\rho) = \{\tilde{\rho} : \|\sqrt{\rho}\sqrt{\tilde{\rho}}\|_1^2 \geqslant 1 - \delta^2\}$, and $\|\cdot\|_1 = \mathrm{Tr}|\cdot|$ is the trace norm. The parameter δ is called a smoothing parameter. For a state ρ^{AB} on a composite system AB, the smooth conditional min-entropy is defined as

$$H^{\delta}_{\min}(A|B)_\rho := -\min_{\sigma^B} D^{\delta}_{\max}(\rho^{AB}\|I^A \otimes \sigma^B), \tag{D2}$$

where the minimization is taken over all states in the subsystem B. The smooth conditional min-entropy satisfies the fully quantum asymptotic equipartition property (FQAEP) [47], i.e.,

$$\lim_{\delta\to 0}\lim_{n\to\infty} \frac{1}{n} H^{\delta}_{\min}(A^n|B^n)_{\rho^{\otimes n}} = H(A|B)_\rho, \tag{D3}$$

where A^n and B^n denote n copies of A and B, respectively, and $H(A|B)_\rho = H(AB)_\rho - H(B)_\rho$ is the conditional entropy based on the von Neumann entropy $H(A)_\tau = -\mathrm{Tr}[\tau^A \log_2 \tau^A]$.

We also use the quantum information variance $V(\rho\|\sigma)$ for the states ρ and σ that satisfy $\mathrm{supp}\,\rho \subseteq \mathrm{supp}\,\sigma$:

$$V(\rho\|\sigma) := \mathrm{Tr}\big[\rho(\log_2\rho - \log_2\sigma)^2\big]. \tag{D4}$$

The following relation is useful for analyzing the second-order asymptotics [48]:

$$D^\delta_{\max}(\rho^{\otimes n}\|\sigma^{\otimes n}) = nD(\rho\|\sigma) - \sqrt{nV(\rho\|\sigma)}\Phi^{-1}(\delta^2) + O(\log n), \tag{D5}$$

where $D(\rho\|\sigma) = \mathrm{Tr}[\rho(\log_2\rho - \log_2\sigma)]$ is the quantum relative entropy, and $\Phi^{-1}(\epsilon) = \max\{z \in \mathbb{R} : \Phi(z) \leqslant \epsilon\}$ with $\Phi(x) = \int_{-\infty}^{x} e^{x^2/2}dx$ is the cumulative distribution function with respect to the normal distribution.

Let us now investigate the logical errors in the block model. Following the convention, we define the logical error Δ_{block} in each block by using the maximally entangled state $|\Phi\rangle$ between the rm logical qubits and a reference system, where $r = k/n$ is the encoding rate. More specifically, when the encoding circuit is U and the decoding operation is $\mathcal{D}$, the logical error against a noisy channel $\mathcal{N}$ is defined by

$$\Delta_{\text{block}}(U, \mathcal{N}, \mathcal{D}) = \frac{1}{2}\big\||\Phi\rangle\langle\Phi| - (\mathrm{id}\otimes(\mathcal{D}\circ\mathcal{N}\circ\mathcal{U}))(|\Phi'\rangle\langle\Phi'|)\|_1, \tag{D6}$$

where id is the identity map on the reference, $\mathcal{U}(\cdot) = U\cdot U^\dagger$ is a unitary channel, and $|\Phi'\rangle = |\Phi\rangle \otimes |0\rangle^{\otimes(1-r)m}$. Taking the minimum over all possible decoding operations $\mathcal{D}$ and taking the average over the encoding circuit U, we define the average logical error $\Delta_{\text{block}}(\mathcal{N})$ in each block. As the total number of blocks is n/m, the total logical error $\Delta(\mathcal{N})$ satisfies

$$\Delta(\mathcal{N}) \leqslant \frac{n}{m}\Delta_{\text{block}}(\mathcal{N}). \tag{D7}$$

We consider below only the IID Pauli noise and omit $\mathcal{N}$ to write $\Delta_{\text{block}}(\mathcal{N})$ and $\Delta(\mathcal{N})$ as Δ_{block} and Δ, respectively.

For stabilizer codes, these quantities are related to the average logical error probabilities since, when decoding fails for stabilizer codes, the resulting state becomes orthogonal to $|\Phi\rangle$. Hence, we have $\Delta = q_L$, where q_L is the probability of an error on at least one logical qubit in the block model. Using the average logical error probability q'_L per logical qubit, it also holds that

$$\Delta_{\text{block}} = 1 - (1 - q'_L)^{rm} = rmq'_L + O((rmq'_L)^2), \tag{D8}$$

$$\Delta = 1 - (1 - q'_L)^k = kq'_L + O((kq'_L)^2). \tag{D9}$$

This also implies that $\Delta = \frac{n}{m}\Delta_{\text{block}}$ to the leading order for stabilizer codes.

In the block model with block size m, the encoding circuit forms a unitary 2-design on each block when the depth of the random Clifford circuit is $O(m)$. Using this fact and the standard decoupling approach [49,50], we obtain an upper bound on the average decoding error Δ_{block} as

$$\Delta_{\text{block}} \leqslant \big(2^{-\frac{m}{2}R(\delta)} + 12\delta\big)^{1/2} \tag{D10}$$

for any $\delta \geqslant 0$, where

$$R(\delta) = \frac{1}{m}H^\delta_{\min}(A^m|E^m)_{\nu^{\otimes m}} - r, \tag{D11}$$

and ν^{AE} is the normalized Choi-Jamiołkowski state of the complementary channel of the Pauli noise on a single qubit. Here, we labeled by A a single physical qubit, and by E an environment of the Pauli noise acting on A. Note that recent studies [51] suggest that Eq. (D10) can be improved to $\Delta_{\text{block}} \leqslant 2^{-mR(\delta)} + 12\delta$. However, we here use Eq. (D10) for simplicity.

Using the FQAEP [Eq. (D3)], it follows that

$$\lim_{\delta\to 0}\lim_{m\to\infty} R(\delta) = H(A|E)_\nu - r \tag{D12}$$

$$= 1 - H(\vec{p}) - r =: R, \tag{D13}$$

where $H(\vec{p})$ is the Shannon entropy of the Pauli error probabilities $\vec{p} = (p_I, p_X, p_Y, p_Z)$. Hence, for sufficiently large n, we have $\Delta_{\text{block}} \leqslant 2^{-\frac{m}{4}R}$ to the first order of m. Since $\Delta = \frac{n}{m}\Delta_{\text{block}}$, it further follows that $\Delta \leqslant \frac{n}{m}2^{-\frac{m}{4}R}$. In particular, when $m = \beta\log_2 n$, the average logical error satisfies

$$\Delta \leqslant \frac{n^{1-\frac{\beta}{4}R}}{\beta\log_2 n}, \tag{D14}$$

to the first order of n. As $\Delta = q_L$ for stabilizer codes, which is the case when the encoding circuit is Clifford, this is exactly the claim in the main text.

To obtain the average logical error Δ up to the second order, we use Eq. (D5). To this end, we rephrase the smooth conditional min-entropy in terms of the smooth max-relative entropy, such as

$$\frac{1}{m}H^\delta_{\min}(A^m|E^m)_{\nu^{\otimes m}} = -\frac{1}{m}\min_{\sigma^{E^m}} D^\delta_{\max}((\nu^{AE})^{\otimes m}\|I^{A^m}\otimes\sigma^{E^m}) \tag{D15}$$

$$\geqslant -\frac{1}{m}D^\delta_{\max}((\nu^{AE})^{\otimes m}\|(I^A\otimes\nu^E)^{\otimes m}). \tag{D16}$$

Using Eq. (D5) and direct calculations such as

$$D(\nu^{AE}\|I^A\otimes\nu^E) = -1 + H(\vec{p}), \tag{D17}$$

$$V(\nu^{AE}\|I^A\otimes\nu^E) = 1 + 2H(\vec{p}) + \sum_j p_j\big(\log_2 p_j\big)^2 \tag{D18}$$

$$=: V(\vec{p}), \tag{D19}$$

we have

$$\frac{1}{m}H^\delta_{\min}(A^m|E^m)_{\nu^{\otimes m}} \geqslant 1 - H(\vec{p}) + \frac{\Phi^{-1}(\delta^2)\sqrt{V(\vec{p})}}{\sqrt{m}} + O\bigg(\frac{\log m}{m}\bigg). \tag{D20}$$

Substituting this into Eq. (D11), we obtain

$$R(\delta) \geqslant R - \frac{\gamma}{\sqrt{m}}, \tag{D21}$$

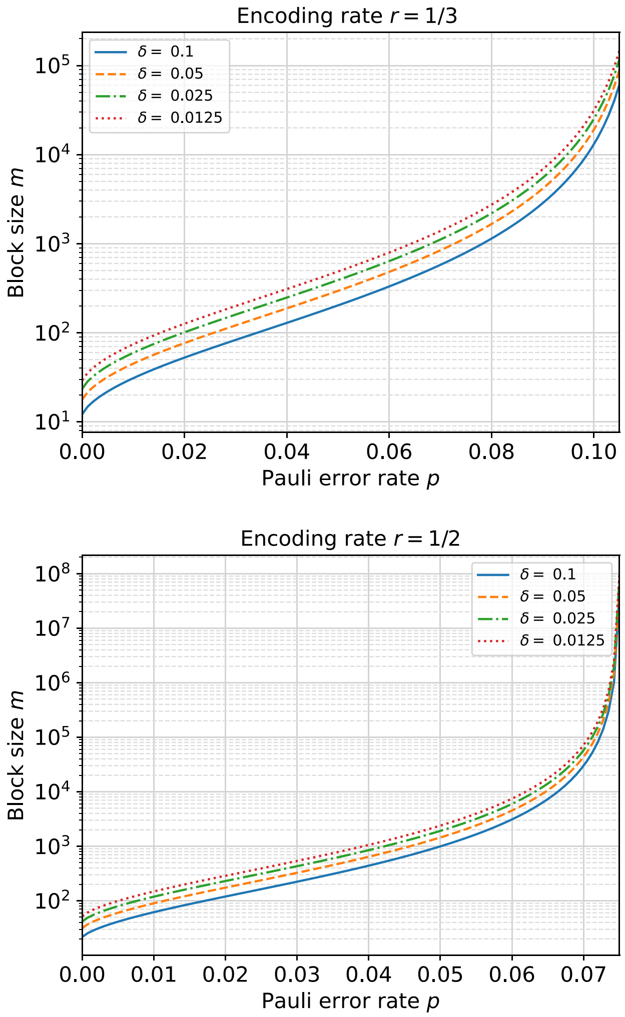

FIG. 12. The right-hand side of Eq. (D24) for the IID Pauli noise with $p_X = p_Y = p_Z = p/3$ and $p_I = 1 - p$, and for the encoding rates $r = 1/3$ (upper figure) and $r = 1/2$ (lower figure). The smoothing parameter δ is set to $0.1,\ 0.05,\ 0.025$, and 0.0125. If the block size m is sufficiently larger than each line, the finite-size effect is negligible. It is clear that the required block sizes grow radically towards the hashing bound.

to the order of $1/\sqrt{m}$, where $\gamma = -\Phi^{-1}(\delta^2)\sqrt{V(\vec{p})}$. Using the relation $\Delta = \frac{n}{m}\Delta_{\text{block}}$, we arrive at the second-order asymptotics of the logical error:

$$\Delta \leqslant \frac{n}{m}\left(2^{-\frac{m}{2}\left(R-\frac{\gamma}{\sqrt{m}}\right)} + 12\delta\right)^{1/2} \tag{D22}$$

for any $\delta \geqslant 0$. Note that, since $\Phi^{-1}(x) < 0$ if $x < 1/2$, $\gamma > 0$ when $\delta < 1/\sqrt{2}$.

When the block size m is given by $m = \beta \log_2 n$, the average logical error is bounded from above by

$$\Delta \lesssim \frac{n^{1-\frac{\beta}{4}R+\frac{\gamma}{4}\sqrt{\frac{\beta}{\log_2 n}}}}{\beta \log_2 n}, \tag{D23}$$

where we ignored δ for simplicity. Recalling $\Delta = q_L$, this provides the second claim in the main text. As $\gamma > 0$ for small δ, the achievable bound to the second order is clearly worse than that to the first order [Eq. (D14)].

From Eq. (D21), the finite-size effect is negligible when $R \gg \gamma/\sqrt{m}$, or equivalently, when the block size m satisfies

$$m \gg \left(\frac{\Phi^{-1}(\delta^2)\sqrt{V(\vec{p})}}{1 - H(\vec{p}) - r}\right)^2. \tag{D24}$$

In Fig. 12, we plot the right-hand side of Eq. (D24) for the Pauli noise with $p_X = p_Y = p_Z = p/3$ and $p_I = 1 - p$ for the encoding rates $r = 1/3$ and $1/2$. It is observed that, when p is small, the block size m of a few tens may suffice for the finite-size effect to be negligible, while m should be drastically large when p approaches the hashing bound. This provides estimations of the block size for the block model with logarithmic block sizes to work.

[1] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina, and C. Monroe, Fault-tolerant control of an error-corrected qubit, Nature (London) **598**, 281 (2021).

[2] M. H. Abobeih, Y. Wang, J. Randall, S. J. H. Loenen, C. E. Bradley, M. Markham, D. J. Twitchena, B. M. Terhal, and T. H. Taminiau, Fault-tolerant operation of a logical qubit in a diamond quantum processor, Nature (London) **606**, 884 (2022).

[3] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, Demonstration of fault-tolerant universal quantum gate operations, Nature (London) **605**, 675 (2022).

[4] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[5] C. Ryan-Anderson *et al.*, Realization of real-time fault-tolerant quantum error correction, Phys. Rev. X **11**, 041058 (2021).

[6] S. Krinner, N. Lacroix, A. Remm, A. D. Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, G. J. Norris, C. K. Andersen, M. Müller, A. Blais, C. Eichler, and A. Wallraff, Realizing repeated quantum error correction in a distance-three surface code, Nature (London) **605**, 669 (2022).

[7] Y. Zhao *et al.*, Realization of an error-correcting surface code with superconducting qubits, Phys. Rev. Lett. **129**, 030501 (2022).

[8] M. Gong *et al.*, Experimental exploration of five-qubit quantum error-correcting code with superconducting qubits, Natl. Sci. Rev. **9**, nwab011 (2021).

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

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

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

[12] D. Gottesman, Fault-tolerant quantum computation with constant overhead, Quantum Inf. Comput. **14**, 1338 (2014).

[13] A. A. Kovalev and L. P. Pryadko, Fault tolerance of quantum low-density parity check codes with sublinear distance scaling, Phys. Rev. A **87**, 020304(R) (2013).

[14] O. Fawzi, A. Grospellier, and A. Leverrier, Constant overhead quantum fault tolerance with quantum expander codes, Commun. ACM **64**, 106 (2021).

[15] P. Panteleev and G. Kalachev, Asymptotically good quantum and locally testable classical LDPC codes, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022 (Association for Computing Machinery, New York, 2022), pp. 375–388.

[16] A. Leverrier and G. Zémor, Quantum Tanner codes, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, New York, 2022).

[17] S. Gu, C. A. Pattison, and E. Tang, An An efficient decoder for a linear distance quantum LDPC code, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (ACM Digital Library, New York, 2023), pp. 919–932.

[18] A. Leverrier and G. Zémor, Efficient decoding up to a constant fraction of the code length for asymptotically good quantum codes, in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, Philadelphia, USA, 2023), pp. 1216–1244.

[19] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, Good quantum LDPC codes with linear time decoders, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (ACM Digital Library, New York, 2023), pp. 905–918.

[20] A. Leverrier and G. Zémor, Decoding quantum tanner codes, IEEE Transactions on Information Theory **69**, 5100 (2023).

[21] H. Yamasaki and M. Koashi, Time-efficient constant-space-overhead fault-tolerant quantum computation, Nat. Phys. **20**, 247 (2024).

[22] D. Gottesman, Fault-tolerant quantum computation with local gates, J. Mod. Opt. **47**, 333 (2000).

[23] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology Pasadena, California, 1997.

[24] M. M. Wilde, *Quantum Information Theory*, 1st ed. (Cambridge University Press, Cambridge, 2013).

[25] Y. Nakata, E. Wakakuwa, and H. Yamasaki, One-shot quantum error correction of classical and quantum information, Phys. Rev. A **104**, 012408 (2021).

[26] D. Sutter, V. B. Scholz, A. Winter, and R. Renner, Approximate degradable quantum channels, IEEE Trans. Inf. Theor. **63**, 7832 (2017).

[27] F. Leditzky, N. Datta, and G. Smith, Useful states and entanglement distillation, IEEE Trans. Inf. Theor. **64**, 4689 (2018).

[28] W. Brown and O. Fawzi, Short random circuits define good quantum error correcting codes, IEEE Int. Symp. Inf. Theor. Proc. 346 (2013).

[29] W. Brown and O. Fawzi, Decoupling with random quantum circuits, Commun. Math. Phys. **340**, 867 (2015).

[30] M. J. Gullans, S. Krastanov, D. A. Huse, L. Jiang, and S. T. Flammia, Quantum coding with low-depth random circuits, Phys. Rev. X **11**, 031066 (2021).

[31] M. Tremblay, G. Duclos-Cianci, and S. Kourtis, Finite-rate sparse quantum codes aplenty, Quantum **7**, 985 (2023).

[32] B. J. Brown, A. Al-Shimary, and J. K. Pachos, Entropic barriers for two-dimensional quantum memories, Phys. Rev. Lett. **112**, 120503 (2014).

[33] M. P. A. Fisher, V. Khemani, A. Nahum, and S. Vijay, Random quantum circuits, Annu. Rev. Condens. Matter Phys. **14**, 335 (2023).

[34] P. Iyer and D. Poulin, Hardness of decoding quantum stabilizer codes, IEEE Trans. Inf. Theor. **61**, 5209 (2015).

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

[36] S. Bravyi, M. Suchara, and A. Vargo, Efficient algorithms for maximum likelihood decoding in the surface code, Phys. Rev. A **90**, 032326 (2014).

[37] C. T. Chubb, General tensor network decoding of 2D Pauli codes, arXiv:2101.04125.

[38] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, Computational complexity of projected entangled pair states, Phys. Rev. Lett. **98**, 140506 (2007).

[39] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011).

[40] A. S. Darmawan and D. Poulin, Linear-time general decoding algorithm for the surface code, Phys. Rev. E **97**, 051302(R) (2018).

[41] C. T. Chubb and S. T. Flammia, Statistical mechanical models for quantum codes with correlated noise, Ann. Henri Poincaré D **8**, 269 (2021).

[42] M. A. Nielsen, A simple formula for the average gate fidelity of a quantum dynamical operation, Phys. Lett. A **303**, 249 (2002).

[43] E. Knill, Quantum computing with realistically noisy devices, Nature (London) **434**, 39 (2005).

[44] E. Knill, Fault-tolerant postselected quantum computation: schemes, arXiv:quant-ph/0402171.

[45] C. Wang, J. Harrington, and J. Preskill, Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory, Ann. Phys. **303**, 31 (2003).

[46] J. Haferkamp, F. Montealegre-Mora, M. Heinrich, J. Eisert, D. Gross, and I. Roth, Efficient unitary designs with a system-size independent number of non-clifford gates, Commun. Math. Phys. **397**, 995 (2022).

[47] M. Tomamichel, R. Colbeck, and R. Renner, A fully quantum asymptotic equipartition property, IEEE Trans. Inf. Theor. **55**, 5840 (2009).

[48] M. Tomamichel and M. Hayashi, A hierarchy of information quantities for finite block length analysis of quantum tasks, IEEE Trans. Inf. Theor. **59**, 7693 (2013).

[49] F. Dupuis, The decoupling approach to quantum information theory, Ph.D. thesis, Université de Montréal, 2009.

[50] F. Dupuis, M. Berta, J. Wullschleger, and R. Renner, One-shot decoupling, Commun. Math. Phys. **328**, 251 (2014).

[51] Y. Nakata, T. Matsuura, and M. Koashi, Constructing quantum decoders based on complementarity principle, arXiv:2210.06661.